Faculty of Computer Science and Information Technology

University of Malaya

Computer Aided Learning Package for Electronic Keyboard

Prepared by : Goh Yeh Hwang

WEK 000079

Supervisor : Assoc. Prof. Dr. Ow Siew Hock

Moderator : Mr. Chiew Thiam Kian

Abstracts

It is a compulsory to fulfill the WXES 3181 and WXES 3182, in order to complete the entire course, Bachelor of Science Computer, University of Malaya.

The objective of internship training is to expose students to the real situation for process of operations, development and management in order to produce a group of high quality students.

I was selected the project title: Computer Aided Learning Package for Electronic Keyboard while the entire courseware teaches fundamental skills to play electronic keyboard, including the elementary sight reading and music theories.

This courseware was designed using Modern Object Oriented approach, Unified Modeling Language and platform independent solution, Java technology.

The key features of this courseware are user friendliness, easy to understand, platform independent, CD-ROM based lessons, add-in tutorial and games with minimal installation requirement.

Acknowledgements

First of all, I would like to express my gratitude to Professor Assoc. Dr. Ow Siew Hock, my supervisor who brought me into this project, Computer Aided Learning Package for Electric Keyboard; Dr. Pan Kok Chang, an experienced music and piano lecturer in Pusat Kebudayaan, University of Malaya. Dr. Pang guides me the corresponding contents and instructions.

Furthermore, I would like to extend my thanks to my moderator, Mr. Chiew Thiam Kian, all my friends who express their friendliness and help to success my project, including students and stuffs from Pusat Kebudayaan, my seniors, and my coursemate.

Table of Contents

List of Figures

Chapter 1 Introduction

1.1 Project Overview

This project is building a CD-ROM based computer aided learning package for the preliminary electronic keyboard. This learning package should be suitable for the users who do not know music, especially for kids.

This learning package is an interactive multimedia system, which applies the integration of text, images, video clips, animations and sounds to produce an attractive learning environment. Accordingly, the knowledge delivers faster and thus increases the effective of overall learning process.

Basically, the lessons of this learning package covered the fundamental electronic keyboard knowledge and playing skills such as basic music theories, staff reading, rhythms, tempo, chords and so on.

Besides the theory lessons, some interesting tutorials, quiz, games and practices are also included in this package to evaluate and help user learn faster. Meanwhile, users can skip or select the desire the entire topic, this will diminish the redundant or boring topics, hence improve the fascinating of the overall learning program.

1.2 Motivation

The traditional ways of learning music are either to attend formal course such as piano tuition, with the professional guidelines from a piano teacher, or self initiative like buy music teaching books and musical instruments to learn on their own. Nevertheless, the traditional ways has limited the chance for everyone to learn music.

The disadvantages of the traditional approaches are:

- Expensive - Attending formal piano tuition normally cost more than a few hundred Ringgit Malaysia per month, which is unaffordable for the low and average income families.
- Slow - Some outstanding students might learn faster than others, but they still need to follow the entire program conducted by the teachers, hence reduce the interest of the particular student.
- Dreary - The plain text learning method (self initiative from the books) is not interactive and sometimes difficult to understand, this will diminish the interest.
- Limited - Not everybody has the chance to attend the music tuition. Meanwhile, self learning users might difficult to learn the concept of tempo, duration and music notation via the explanations without the exposure of the appropriate sounds and songs.

The outcome of this project is to produce a CD-ROM based electronic learning package suitable for all ages, in an easy to understand approach and finally helps users to build up the skills to play electronic keyboard, runs under Microsoft Windows operating system.

1.3 Project Scope

The scopes of this project covers certain topics, these are report writing, system analysis, system design, system development and testing to turn out the final product. The final product is a CD-ROM based computer program, caters the basic skills and knowledge needed to play electronic keyboard. The outcome of this package consists of following modules:

- Introduction - Bring in the facts about electronic keyboard such as instrument history, family, types, care, and also introduce the significant pianists.
- Lessons - The lessons module teaches basic music theories and practices to play piano, with embedded some masterpiece from classical or popular piano works.
- Games - Allows users to play and learn for the entire topic they have gone through, intends to improve user's understanding and memory.
- Virtual keyboard – A electronic keyboard embedded in the package that allows user to manipulate the electronic keyboard with additional functions.

At any point, the contents of the entire lesson, such as notes, music works, quiz, exercises, and chords can be printed out for off line practice.

1.4 Project Objective

Achieving the objectives determine the success of the entire project. With these objectives clearly stated out, the research studies, design and development works can be aligned towards the desire success the project easily.

Below are the objectives of doing this project:

- Develop a complete and reliable interactive multimedia application follow the modern software engineering practices.
- Design and develop an Object Oriented application for CAL.
- To help individual learn music in more interactive, easier and cheaper manner.

1.5 Development Process

The courseware development process approaches the object-oriented based development model uses unified modeling languages (UML). The UML identify five workflows cut across the entire development phases, where every workflow is set of activities that needs to perform.

1.5.1 Requirement

The requirement workflow captures the functional requirements of the desire system and results the use case model. The use case model is the foundation model for all other development work.

1.5.2 Analysis

The analysis workflow refines and structures the functional requirement captured within the use case model and aims to builds the analysis model.

1.5.3 Design

The design model which describes the physical realizations of the use cases from the use case model and analysis model is built in the design workflow. The design model is the abstraction of the implementation model. The design workflow also builds the deployment model to describe the physical organization of the system.

1.5.4 Implementation

The implementation model from the implementation workflow describes how the elements of the design model are packaged into software element.

1.5.5 Test

Lastly, the primary activities of the test workflow are aimed at building the test model. The test model describes how integration and system tests will carry out on the executable components from the implementation model.

1.6 Multimedia Instruction Project Schedule

ID	Task Name	Start	Duration
1	Project Planning	6/10/2002	25d
2	Literature review	6/25/2002	28d
3	Study Existing System	7/7/2002	10d
4	Requirement analysis	7/18/2002	30d
5	System Design	8/12/2002	50d
6	System Development	9/20/2002	60d
7	System Testing	10/27/2002	40d
8	System Review and Evaluation	12/6/2002	12d
9	Documentation	6/10/2002	200d

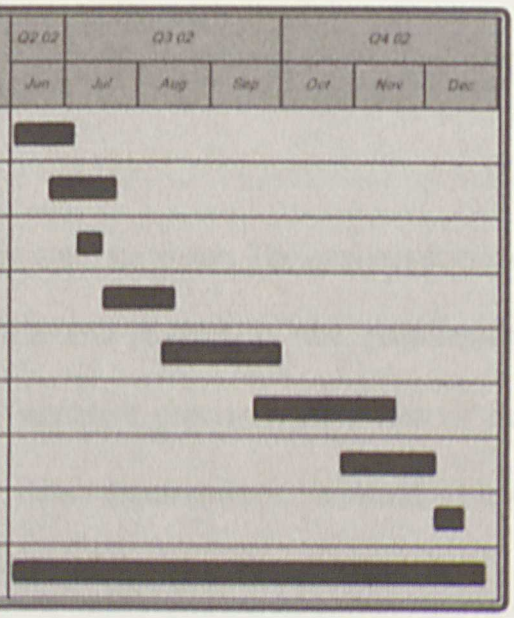

Figure 1.1 Overall Project Schedule

1.7 Report Layout

1.7.1 Chapter 1: Introduction

Generally introduce the system to be built, such as motivations, objectives, scopes and project planning. This chapter is important because the clarity of project objectives and planning are the key factors toward the success.

1.7.2 Chapter 2: Literature Review

Researches, comparisons, studies and collecting the related technologies and information to build the electronic keyboard multimedia teaching modules and embedded program, for instance, the tools, database, APIs, programming languages, multimedia technologies, MIDI sequencing, music notation and etceteras.

1.7.3 Chapter 3: Methodology

This chapter is describes the methodology selected to perform the design and development process. Unified Modeling Language from OMG is selected for this project, and the very details of the selected modeling method is describes in this chapter.

1.7.4 Chapter 4: System Analysis

This chapter is documented the finding from the analysis phase. The analysis focused on the system requirement analysis and the end-product is the requirement specification. The requirement analysis is to establish precisely the needs of the system, included functional requirements, data requirements, constrains and guidelines. [1]

1.7.5 Chapter 5: System Design

In an object-oriented approach, system design will emphasis on providing an architectural model of system for soon implementation. In investigate the primary goals of design, the reusable software normally presupposes quality, hence both quality and maintainability are vital design goals. In Mayer (1988) it is estimated the maintenance accounts for 70% of the actual cost of software, thus the documentation for software design is significant. The outcomes of design must serve both justification and clarification of the actual implementation. [2]

1.7.6 Chapter 6: Implementation

Errors may occur during the implementations as well as later the system was distributed. Apart from the errors correctness, others maintenance activities such as adaptive maintenance may need to carry on. The documentation for implementation specified all the details needed for system to run.

1.7.7 Chapter 7: System Testing

The system testing is important to make sure the final product is achieved the goal of development. Some testing techniques are introducing in the software engineering such as up-down or Sardine mechanism. This chapter defines the selected approach and then models the suitable test plan. The test plan is important to make sure the programs combination work properly, and be able to capture the errors, faults and mistakes.

1.7.8 Chapter 8: System Evaluation

The evaluation has two, formative and summative. The formative evaluation activity involved individual and small group trials, to try out and revise the course material. This is conducted while the course is under development. The summative evaluation conducted while the course is in regular use, to measure the effectiveness of the entire course for the related subject.

1.8 Summary

A lot of multimedia programs have emerged to teach various forms of musical instrument such as piano, guitar and harmonica, every one with their own teaching technique. However, there is still a lot of room left for improvement - no one integrated a build-in software to allow user further explore to other music pieces easily:

listen - how would the music notation sounds like.

see - how should the music notation being play.

evaluate - how good can user play the songs.

Chapter 2 Literature Review

2.1 Overview

Several resources are being used to produce the outcomes of the review, include Internet, newspaper, magazines, journals, books, and senior students' WXES 3181 final reports. Some fact-finding techniques like interviewing, questionnaires and observation was done to collect fact and information pertaining to the system requirements.

At the end of review, the most suitable tool will be selected; the system requirements will be refined and the outcomes of this literature review will then proceed to system design.

2.2 Computer Aided Learning

Computer aided learning (CAL) describes a learning environment where a computer program is integrated to teach a particular subject, after re-assessment of current teaching method. The CAL is not only a single computer program, it can interpret as art of education strategy devised to teach a particular subject. [3]

2.2.1 Interactive Multimedia

Actually, the trainers have used multimedia training for many years. The term multimedia refers to combination of more than one medium, for example, audiotape,

combined with slides and filmstrip was widely used for training since long time ago. [3]

Today, multimedia is used to teach cognitive objectives, especially involves in the training to recognize and verify the applicable visual stimuli and audio stimuli. It is also a good tool to simulate the real-world conditions as closely as possible for more advance training.

2.2.2 *Advantages of interactive multimedia*

- directly support on-job performance
- can access to a variety of media and other resources easily
- possibility to access the specific information wanted by learner quickly
- provides the individual and self-pace instruction for individual students
- allowing the instructor to concentrate on instruction-related tasks and pay more attention on those students who may having problems
- well simulate the real situations

2.2.3 *Disadvantages of instruction multimedia*

- does not teach or instruct
- may not tie directly to the particular learning objective
- the access to the information is uncontrolled, student might access to the information depends on their interest and motivation
- development of lessons requires skills may not be available
- generic multimedia lessons may not meet the users needs

2.3 Electronic Keyboard

Keyboard instrument comprise three main families differentiated by their means if tone of production. String keyboard include those wherein stretched string are sounded by plucking striking or bowing. Organs constitute a second family, involving whistle like flue pipes and that incorporate vibrating reeds. The third category, in which hammers hit resonant solid bodies such as bells, includes tower carillons and keyed glockenspiels. [4]

Electronic keyboard is normally taught by piano teacher. The fee to attend the music tuition for absolutely beginner is between RM 30 – RM 60 per half an hour. The tuition is recommended to be taught in one teacher to one pupil manner. Some music schools also provide the class room style tuition but the target always for kids below teenage age.

2.3.1 Compares two teaching piano CD-ROM

Two existing courseware was selected for the review purpose. Teach Yourself to Play Piano from Alfred and Easy Piano from ARC Media is two different types of product available and popular in the market.

2.3.2 The Teach Yourself to Play Piano from Alfred

This CD-ROM title is totally a CD-ROM based learning package and it supports Macintosh Operating System. This learning package has some advance features such as a virtual keyboard, where the key on the keyboard shown will highlighted when the note of the song is playing, and performance recording with microphone or MIDI devices.

This CD-ROM title is built with the Macromedia Director software, one of the popular tools for simple games and interactive multimedia development.

Some negative things about this package is the User Interface to dark and the background Music is not attracting, it is horrible and scary.

2.3.3 The Easy Piano from ARC Media

The Easy Piano is one of the series products from ARC for some musical instrument learning lessons such as guitar, bass and piano.

This courseware is requires installation and the design tool also is director.

Table 2.1 Comparing Two Piano Teaching CD-ROM Titles

Features	Teach Yourself to Play Piano from Alfred	Easy Piano from ARC Media
Installation needed	Yes	No
CD-ROM Based	No	Yes
Required component	QuickTime	-
Step-By-Step Lessons	No	Yes
Music Theories	Weak	Good
Sight Reading	Yes	Yes
Finger Aerobics	Yes	No
Video Clips/Animation	No	Yes
Virtual Keyboard	Yes	No
Tempo Control	Yes	No
Performance recording	Audio/MIDI	No
Printing Support	No	Yes
Upgradeable	Yes	No
Supported OS	Window95/NT 4.0 or higher	Windows, Macintosh

2.4 Programming Languages

2.4.1 *Microsoft Visual C++ 6.0*

Microsoft Visual C++ provides a graphical integrated development environment for creating two types of C++ programs. The first type, console application, is a traditional character-based program runs under non-windows based operating system, such as DOS or UNIX. The second type, event-driven programs which waits user activates a control such as a button, and responds to the selected control. [5]

The Microsoft Visual C++ package includes the C/ C++ compiler and many others files and tools necessary to compile and link windows-based programs. The Visual C++ Developer Studio is an IDE to edit source code; create resources such as icons and dialog boxes; to edit, compile, run, and debug the programs.

Microsoft Corporation provides plenty of resources, technical supports and documentation for their products. The Microsoft Developer Network is a program that provides developers with frequently updated CD-ROMs containing much of what they need to be on the cutting edge of Windows development.

2.4.2 Microsoft Visual Basic 6.0

Visual Basic is Microsoft's RAD tool for Windows programmers. Visual Basic uses a graphical, forms-based approach to application development. The typical way to write a program in Visual Basic is to create a form, drag and drop components onto the form, set properties for the form and its components, and add application-specific code to handle events. It sounds simple, but you can write some very powerful applications this way.

Visual Basic hides most of the gory details of Windows-based programming and focuses on application-specific functionality. Every service used in a Visual Basic program is provided by an ActiveX component. Using any component in Visual Basic, simply a matter of learning how its object models - its properties, methods, and events. [6]

2.4.3 Java

The Java™ platform realized the idea that the same software should be able to run on any different kinds of computers, consumer gadgets, and other devices. Any Java application can easily be delivered over the Internet, or any network, without operating system or hardware platform compatibility issues. [7]

Java is a programming language for creating program that needs maximum flexibility to interact with environments out of programmer's control. Java application needs Java Virtual Machine (JVM) to interpret the program class file, which was compiled by the programmer.

Advantages of Java:

- Cost – Java is free in sense of usage and intellectual property right, it is Open Architecture Licensed
- Portability – platform independent
- Reusability

Disadvantages of Java:

- Size – Java code requires JVM and thus increases the program size.
- Classpath difficulties – confusing, prone to typos and hard to manage.

Documentation – Documentation for Java is very hard to find hence burden the learning efforts.

2.4.4 Comparing the three languages:

Table 2.2 Comparing Three Programming Languages

Features	Visual C++	Visual Basic	Java
Platform independent solution	no	no	Yes
Speed	fast	fast	Slow
Support MIDI sequencing	yes	yes	Yes
Documentation	good	plenty	Poor
Popularity	popular	popular	Popular
Support Multimedia	yes	yes	Yes
Cost	Expensive	Expensive	Free
Object oriented	yes	no	Yes
Ease of use	intermediate	easy	Hard

2.5 Technologies

2.5.1 Multimedia

Multimedia literally means "more than one medium". Therefore, any combination one or more of medium such as text, graphics, sound, motion, animation, video and interactivity is considered multimedia. [8]

Multimedia has the following advantages:

- Interactivity improve learning process
- Able to illustrate some topics better than static text or diagrams
- Distributable via the web and CD-ROM
- Exciting, unique and attention-grabbing

2.5.2 *Compact Disk*

Compact discs (CD) are stored data in beams and pits. To retrieve the data stored in CD, laser light is reflected off the surface onto a photo detection device that interprets the presence or absence of light as 1s and 0s (binary code). To write data, a pinpoint beam of light is used to burn pits into the CD's surface. To produce large amount of discs faster, a glass master is produced and used to stamp the rest of the discs. The CDs can produce once and no further writing while the data stored not yet reached the total storage of the media.

These are several standard disc formats in the market, such as Compact Disc Digital Audio, CD-ROM, CD-i, Photo CD, Video CD, CD-R every one with their own characteristics. The file system of a CD might refer ISO9660, HFS or hybrids. The ISO 9660 file system most possible only supported under PC while HFS (Hierarchical file system) does support the Macintosh interface convention. The hybrid HFS-ISO disc contains two separate versions with half of the disc each, one for each platform. [9]

2.5.3 *MIDI*

MIDI stands for musical instrument digital interface, was standardized in 1983. The MIDI consists of both a simple hardware interface, and a transmission protocol to be used in controlling any electronic musical device. MIDI is useful to remotely or automatically control musical instruments for musicians, regardless the particular manufacturer's product, or particular type of instrument is being used. [10]

The MIDI connectors on an instrument are female 5-pin DIN jacks. There are separate jacks for incoming MIDI signals and outgoing MIDI signals. The MIDI cable is used to connect MIDI jacks of various instruments together, and it is male 5-pin DIN connectors.

The Standard MIDI File (SMF) is a file format used to store MIDI data. The format was generic so all the sequencers can read the most important data in MIDI messages. The MIDI file format saves data in chunk[1] and binary data. There are 3 different types of MIDI format: type 0's MIDI contains a single track; type1's MIDI contains one or more simultaneous tracks, each on a single midi channel with possibility of maximum 16 channels; type 2's MIDI file contains one or more sequentially independent single-track patterns. [11]

2.5.4 Compression and decompression

Most of computer data files are fairly redundant, they listed the same information over and over again, and the file-compression simply gets rid of the redundancy. A file-compression algorithm lists that information once and then refers back to it whenever it appears in the original file, Instead of repeating to list a piece of information. The decompress algorithm reads the file and replaces the tokens with the original data, inflating the file back to its original size.

Most compression programs use a variation of the LZ adaptive dictionary-based algorithm to shrink files. "LZ" refers to Lempel and Ziv, the algorithm's creators, and "dictionary" refers to the method of cataloging pieces of data.

[12]

2.5.5 *Encryption and decryption*

A lot of information that we don't want other people to know, such as:

- Credit-card information
- Social Security numbers
- Private correspondence
- Personal details
- Sensitive company information
- Bank-account information

Computer encryption is based on the science of cryptography. There are two major categories on computer encryption:

- Symmetric-key encryption
- Public-key encryption

2.5.5.1 Symmetric Key

Symmetric-key encryption is like a secret code that each of the two computers must know in order to decode the information. For example, a coded message which each letter is substituted with the letter that is two down from it in the alphabet, so "A" becomes "C," and "B" becomes "D". Anyone else who sees the message will see only nonsense without the key “shift by 2”.

2.5.5.2 Public Key

Public-key encryption uses a combination of a private key and a public key. User knows the private key, while the public key is given by other computer that user

wants to communicate securely with. To decode an encrypted message, both public key, and user's own private key are required. [13]

2.6 Multimedia Design Tools

2.6.1 *Cakewalk Home Studio 2002*

Cakewalk is a well known software company, expert in producing the music related product such as MIDI synthesizer, Visual Jokebox, Virtual piano and etceteras. The Cakewalk Home Studio 2002 is an audio sound edit application, yet it also a MIDI synthesizer for electronic keyboard.

The Cakewalk Home Studio 2002 is excellent program to edit and mixing the computer sound. It supports wave, apple AIFF, MPEG, Windows Media and Next/Sun audio files. The program manipulates the audio files and exports to wave, MP3, Real Audio and Windows Media Advance Streaming format. The only limitation of this program is it cannot convert the MIDI file to audio such as wave.

2.6.2 Sonic Foundry Acid Pro 3.0

ACID Pro 3.0 is an award-winning loop-based music creation tool to produce original, royalty-free music. Use ACID to create songs, remix tracks, develop music beds, score videos, and develop music for web sites and Flash animations. This revolutionary tool is intuitive and easy to use. It works with hundreds of loops in multiple genres, uses unlimited tracks of audio, imports complete songs, even MP3s and also built-in support for MIDI, video scoring. The output can be saved to a

number of audio and video file formats, including WAV, WMA, RM, AVI, and MP3.

2.6.3 Adobe Photoshop 7.0

Adobe Photoshop is a very popular professional image editing software to create the excellent quality images for print, the Web, and anywhere else.

Some popular features of Adobe Photoshop 7.0 are:

- Sophisticated painting tools including brushes that simulate natural media
- Industry-standard pen tool for precision drawing
- Pattern Maker to generate background textures automatically
- Tools to define and edit slices directly in Photoshop
- Transparency controls including dithered transparency for edges that blend into any background
- Use channels for high-resolution display in critical areas
- Vector output options for resolution-independent text and shapes
- Instant GIF animations from layered Photoshop and Adobe Illustrator® files

[14]

2.6.4 Macromedia Flash MX

Macromedia Flash MX is used to create streaming vector animation, lightweight video, and Rich Internet Applications, both for devices and for the web. Flash MX is complete with a player that can even fit into PDAs and latest-generation phone/organizers from Nokia.

2.6.5 *Macromedia Director 8.0*

Director Shockwave Studio is popular in creating extensible multimedia content that utilizes long form (or streaming) video and deployable across multiple mediums, especially learning applications or advanced 3D games.

2.7 Development Tools

2.7.1 *Microsoft Win32 API*

Microsoft Win32® Application Programming Interface (API) encompasses all the function calls that the application can make of an operation system with the associated data types and structures. The Win32 API is the most important component for almost every 32-bits program written for Windows now day. [15]

2.7.2 *COM Components*

COM is acronym of Component Object Model. It defines a language-independent binary standard for building object that is interoperating in any programming language. The COM interface is defined with the interface definition language. One way to convert these definitions into binary object is using Microsoft Interface Definition Language. The binary interface definitions allow runtime reuse and implementation of the object classes. [16]

2.7.3 *Microsoft Foundation Classes*

Microsoft Foundation Classes Library (MFC) is the C++ class library provided by Microsoft itself, to place an Object-oriented wrapper around the Window APIs. The MFC is also an application framework, which helps in define the structure of an

application, handle routine tasks and also provides abstractions to what the Windows API has offered. [17]

2.7.4 DirectX 8.1 API

DirectX is an interface between programs and the drivers that run graphics, sound and other computer hardware, so the application written for DirectX are flexible for any sound card or graphics adapter might be installed in the end-user's machine.

DirectX plays a role in many functions, including 3D rendering, MIDI sequencing, music and video playback, still and motion capture, mouse interfaces and lots more. [18]

2.7.5 Java JDBC

The "Write Once, Run Anywhere™" Java™ 2 Platform is a safe, flexible, and complete cross-platform solution for developing robust Java applications for the Internet and corporate intranets. The open and extensible Java Platform APIs are a set of essential interfaces that enable developers to build their Java applications and applets. The Java 2 Platform provides uniform, industry-standard, seamless connectivity and interoperability with enterprise information assets.

The JDBC™ API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The JDBC API provides a call-level API for SQL-based database access. JDBC technology allows developers using the Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data. [19]

2.8 Databases

2.8.1 Microsoft Office Access version 2002

Microsoft Access version 2002 delivers powerful tools for managing and analyzing data. With XML support, Microsoft Access now can create sophisticated enterprise-wide database solutions that integrate easily with the Web. New features like Microsoft PivotTable® and Microsoft PivotChart® greatly improve data analysis capabilities.

2.8.2 MySQL

MySQL is a , open-source database developed by Sweden's MySQL AB. Aiming for reliability and speed instead of bells and whistles, MySQL can do the job, provided you don't mind putting some work into daily administration and database development. The server garnered fine results in our performance test, trailing Oracle9i by only a small margin.

MySQL's lean-and-mean philosophy eschews a number of features that are standard fare in the other database packages we evaluated. It does not support stored procedures and triggers, nor does it offer built-in support for XML, OLAP, or constraints. But it includes innovations, like the ability to cache queries.

As with earlier versions, the beauty of MySQL is its fundamental simplicity. You administer the database using over a dozen command-line utilities, the most important of which is the *mysql* utility, an interactive shell for controlling and querying a database. In our testing, we found the utilities to be efficient in the best spirit of Linux, the platform on which MySQL was first developed. We had no

trouble importing and exporting millions of records to get started with our online store database. Besides running SQL scripts in batch mode, the utilities let you "dump" the contents of a database in text format.

While other database products are moving toward putting more business logic into stored procedures (or even into Java business objects) that reside *inside* the database tier, MySQL avoids this approach entirely, preferring to stay light. Though it lacks many features enterprise administrators and database developers have come to expect, MySQL has an energetic and loyal following, not to mention a hard-earned reputation for reliability. Though it won't fit every need, this cheap and fast alternative can be a great choice if transaction processing isn't paramount to your project. [20]

2.9 Conclusion

2.9.1 Java as Platform Independent Solution

The programming language selected upon to develop this computer aided learning package is Java from sun Microsystems, including the lessons module to represent multimedia instruction and games module.

I chose Java as my programming language for this project because:

- Complete multimedia support
- Platform independent
- No need to register any add-in component like what Microsoft ActiveX controls do

- Java is the latest and growing technology, especially in multimedia, image editing and mobile phone programming
- Macintosh is the most popular multimedia environment, hence the target user might comes from Mac user.

Although Java provided the platform independent solution, but because it is a hybrid language, where the compiled codes need to be interpreted last minute before execution, hence increase the size and time.

2.9.2 MySQL as Portable Database

I chose MySQL as the database because I concerned the Microsoft Access is not popular database among the Macintosh users. The Microsoft Access for Macintosh version does not as popular as other Office tools such as Word, PowerPoint and Excel.

MySQL is a stable and free database and support most environment now a day such as Linux, UNIX, Microsoft Windows and Macintosh, so it is the best choice ever. Furthermore, it is free!

Chapter 3 Methodology

3.1 Overview

A software development project normally does not start in complete isolation, hence it is not written from scratch. Software might need to interface with other existing system, or it used existing subroutine libraries, or built upon an existing framework, and so on. In general, a software development project product a set of programs providing the needed functions.

A methodology formally defines the process programmers use to gather requirements, analyze the requirements, and design an application that meets the requirements. There are many methodologies, each differing in some way or ways from the others, and one methodology may be better than another for the particular project.

3.2 Software Life Cycle

The phase development method was used in order to control progress of entire project. Phase development established a number of clearly identifiable milestones between the start and end of the project. The milestones identified in a software development project corresponding to the time when certain documents become available, for example:

- after the requirement engineering, there is a requirement specification.
- after testing has been completed, there is a test report.

Basically, the different between phases is the aims laid behind. According to the UML approach, the software life cycle contains four phases, which are inception, elaboration, construction and transition.

3.2.1 *Inception*

This inception phase establishes the case for the multimedia learning package. The tasks required to perform are as below:

- defining the scopes of the system.
- outlining the candidate architecture, including number of aspects, such as static elements, dynamic elements, how those elements work together, and issues the performance, scalability, reuse, and economic and technological constrains.

3.2.2 *Elaboration*

The primary goal of this phase is to ascertain the ability to build the multimedia-learning package accordingly to the financial constrains, schedule constrains, and other constrains that the development project faces. The primary tasks carry out in this phase is:

- Capturing the remaining functional requirements.
- Expanding the candidate architecture into a full architecture baseline.

3.2.3 *Construction*

The primary goal of the construction phase is to build the learning package that is capable to operate and fulfill the needs. The major task involves within this phase is building the system.

3.2.4 *Transition*

In this phase, the tasks need to perform are correcting defects and modify to correct previously unidentified problems. The major milestone for this phase is roll out the fully functional system (Submit the WKES 3181/3182 end result). [21]

3.3 Unified Modeling Language

Unified Modeling Language (UML) is the international standard notation for object-oriented analysis and design. It is defined by the Object Management Group (OMG) and is currently at release 1.4. The OMG specification states:

> *"The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components."*

[22]

UML also helps to specify, visualize, and document models of software systems, including the structure and design, in a way that meets all of these requirements.

UML defines twelve types of diagrams, divided into three categories:

- **Structural Diagrams** represent static application structure: Class Diagram, Object Diagram, Component Diagram, and Deployment Diagram.
- **Behavior Diagrams** represent different aspects of dynamic behavior: Use Case Diagram, Sequence Diagram, Activity Diagram, Collaboration Diagram, and State Chart Diagram.
- **Model Management Diagrams** represent ways to organize and manage the application modules include Packages, Subsystems, and Models.

3.3.1 *Views of System*

UML defines five views of a system's architecture, where each reveals from a different perspective and hide the aspects that are not concern to the viewer. The use case view concerns the scenarios user interact with the learning package; design view focuses on the problems and solution for the entire project; process view caters those aspect involve timing and flow of control of interactive multimedia software; implementation view focuses on how to assemble things to form the physic system; deployment view focuses on the geographic distribution of the various software elements on hardware and other physical elements, but it is not practical for this WKES 3181/3182 project.

3.3.2 *Structural Diagrams*

3.3.2.1 Class Diagram

Class is a standard UML construct used to detail the pattern from which objects will be produced at run-time. The class modeling is an important object-oriented

development and design technique. The class diagram expresses both the persistent state of the system and the behavior of the system.

Well-designed object-oriented system shall prevent direct access to class attributes and offers services which manipulate attributes on behalf of the caller. The data encapsulation ensures data updates are only done in one place and according to specific rules, hence reduces maintenance burden of code for large systems.

3.3.2.2 Object diagram

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

3.3.2.3 Component diagram

Component diagrams describe the organization of physical software components, including source code, run-time (binary) code, and executables.

3.3.2.4 Deployment diagram

Deployment diagrams depict the physical resources in a system, including nodes, components, and connections.

3.3.3 *Behavior Diagrams*

3.3.3.1 Use Case Diagram

According to the UML specification, version 1.4 by OMG, a use case diagram shows the relationship among use cases within a system or other semantic entity and their

actors [23]. Use case diagram is designed to express all of the things that can be done by a system, and by all of the users.

3.3.3.2 *Sequence diagram*

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

3.3.3.3 Activity diagram

Activity diagrams demonstrate the dynamic nature of a system by model the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

3.3.3.4 Collaboration diagram

Collaboration diagrams illustrate interactions between objects as a series of sequenced messages. Collaboration diagrams are presenting both the static structure and the dynamic behavior of the system.

3.3.3.5 Statechart diagram

Statechart diagrams show the dynamic behavior of a system in react to external stimuli. Statechart diagrams are useful in modeling reactive objects whose states are triggered by specific events.

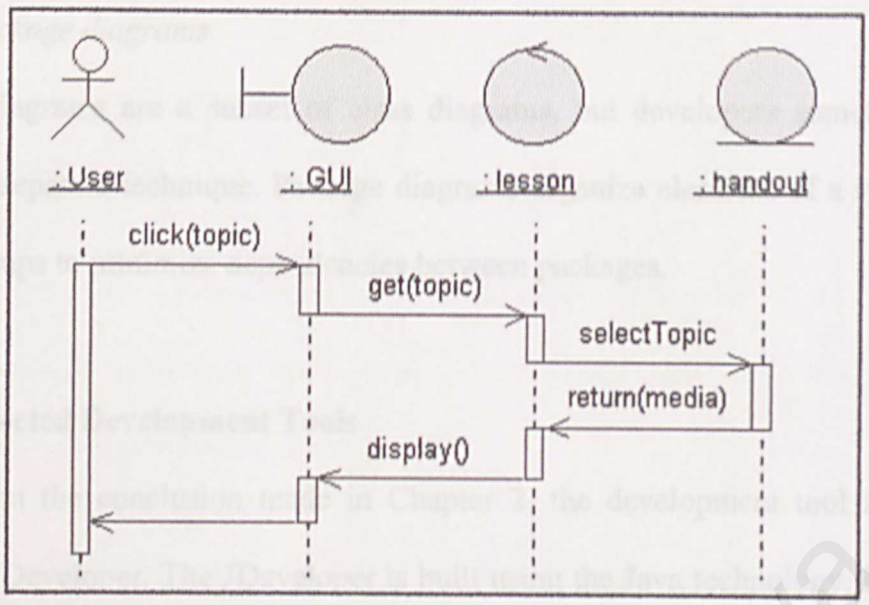

Figure 4.2 Sequence Diagram for Topic Study

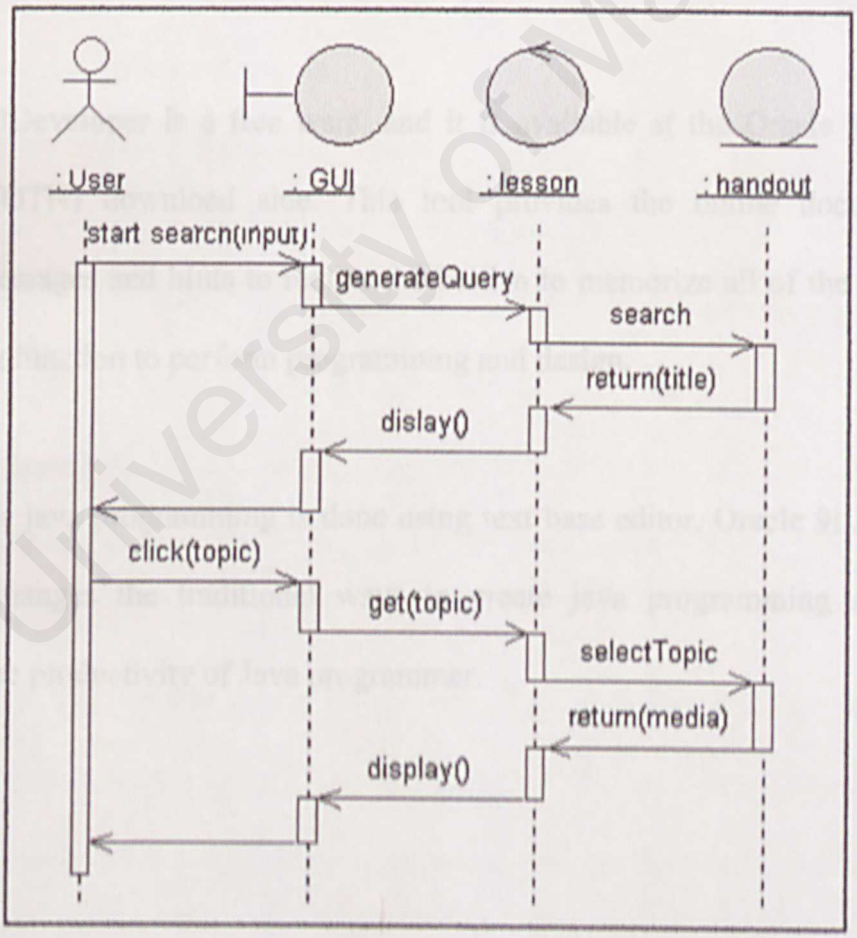

Figure 4.3 Sequence Diagram Illustrated User Performs Search Topic

3.3.4 *Model Management Diagrams*

3.3.4.1 Package diagrams

Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.

3.4 Selected Development Tools

Derive from the conclusion made in Chapter 2, the development tool selected is Oracle 9i JDeveloper. The JDeveloper is built using the Java technology. It provides comprehensive user interfaces hence it also is an excellent integrated development environment (IDE).

Oracle 9i JDeveloper is a free ware, and it is available at the Oracle Technology Network (OTN) download side. This tool provides the online documentation, compile messages and hints to reduce the burden to memorize all of the syntax and API calling function to perform programming and design.

Usually, the java programming is done using text base editor, Oracle 9i JDeveloper IDE has changes the traditional ways to create java programming and highly increases the productivity of Java programmer.

Chapter 4 System Analysis

4.1 Overview

System analysis defines finding of requirement engineering such as functional and non-functional requirements specification, verification and validation, software requirements and hardware requirements.

4.2 Target Users

The target audiences for this courseware package were defined:

- children – from age 1 through 101, especially those in primary standard.
- who wants to learn music or
- wanting to play electronic keyboard
- parents who wish their kid knows to play electronic keyboard
- teacher who wants to improve the effectively of elementary electronic keyboard teaching lessons for their students
- beginner with limited music exposure or without any music training before but wanted to play music instrument like electronic keyboard

To learn how to play any music instrument, practice is the key, but repetition is boring and sometime it is too difficult to understand. This courseware was written in such a way that the user simply plays a game and through the course of play they will learn and understand.

4.3 Functional requirements

This courseware is designed to have two main functions embedded into the six modules of the entire package:

- Displays the contents in the interactive way such as uses hypermedia approach to attract user's interest to explore, allows user to bookmark or look up for entire topic, hence increase the ease of use.
- Games is the recommended ways to help user grasps and memorizes the introduced lessons, while analyze the performance statistics from generated charts or figures help in evaluating user's understanding.

4.3.1 Courseware modules

User selects one of the options provided in the main menu. Subsequent actions are described in the following sections 4.3.1.1 through 4.3.1.5. At any point, user has the option to return to the main menu or even shut down the program.

4.3.1.1 Introduction

This module introduces the collected facts about electronic keyboard, piano and some significant pianist. The user may choose to read the topics chapter by chapter or search topics that matched the input given.

4.3.1.2 Lessons

When the user starts the program, he/she can continue with the last visited topic, with the precondition that the courseware program is run on the same computer.

Several operations are provided in this module. At any point, user can select one of the operations such as reverses to previous or forward to next topic in the contents tray or he/she has visited, skip to the topic he/she has book-marked, search a topic to study, bookmark and print the entire active topic with the preset layout.

4.3.1.3 Games

A game was designed for this learning package, the **Music Hunter**. No installation require, the game can directly play from the CD-ROM, while user can play with the real electronic keyboard which was connected to the sound card or with mouse and click on the key on the screen.

Music Hunter is a computer game where player needs to verify the falling notes randomly picked by computer, and returns the appropriate answer within the time interval to avoid the game terminate. This game introduces the keys on the electronics and helps user to memorize every key on the keyboard in the faster and more interesting manner.

4.4 Functional Description

This learning package is divided into two groups of application: the lessons teaching modules and interactive program such as games and quiz. The first group is only manipulates the static contents with the interactive manner from the CD-ROM so no installation is required. The second group needs upgradeable database thus needs a minor installation.

4.4.1 Use Case Diagrams

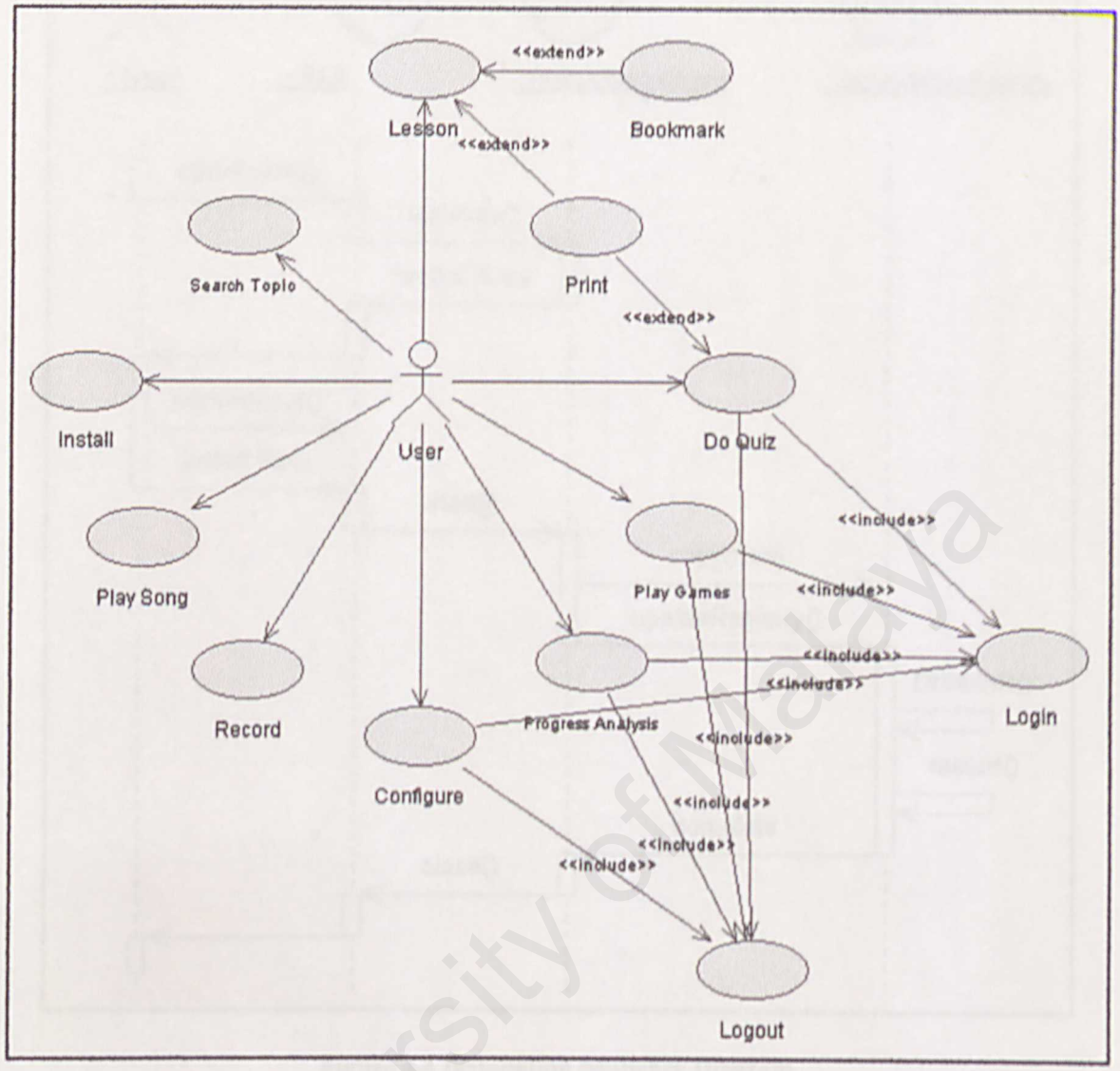

Figure 4.1 Main Use Case Diagram

Figure 4.1 shows the use case diagram to illustrate all possible operations where the targeted user can perform with the entire application.

4.4.2 Sequence Diagrams

User is allows to quit, skip, reverse, forward, bookmark or even print the entire topic or music piece at any point, hence the main control will always with use to interrupt the system to perform the required function. This shows in figures 4.2 and 4.3.

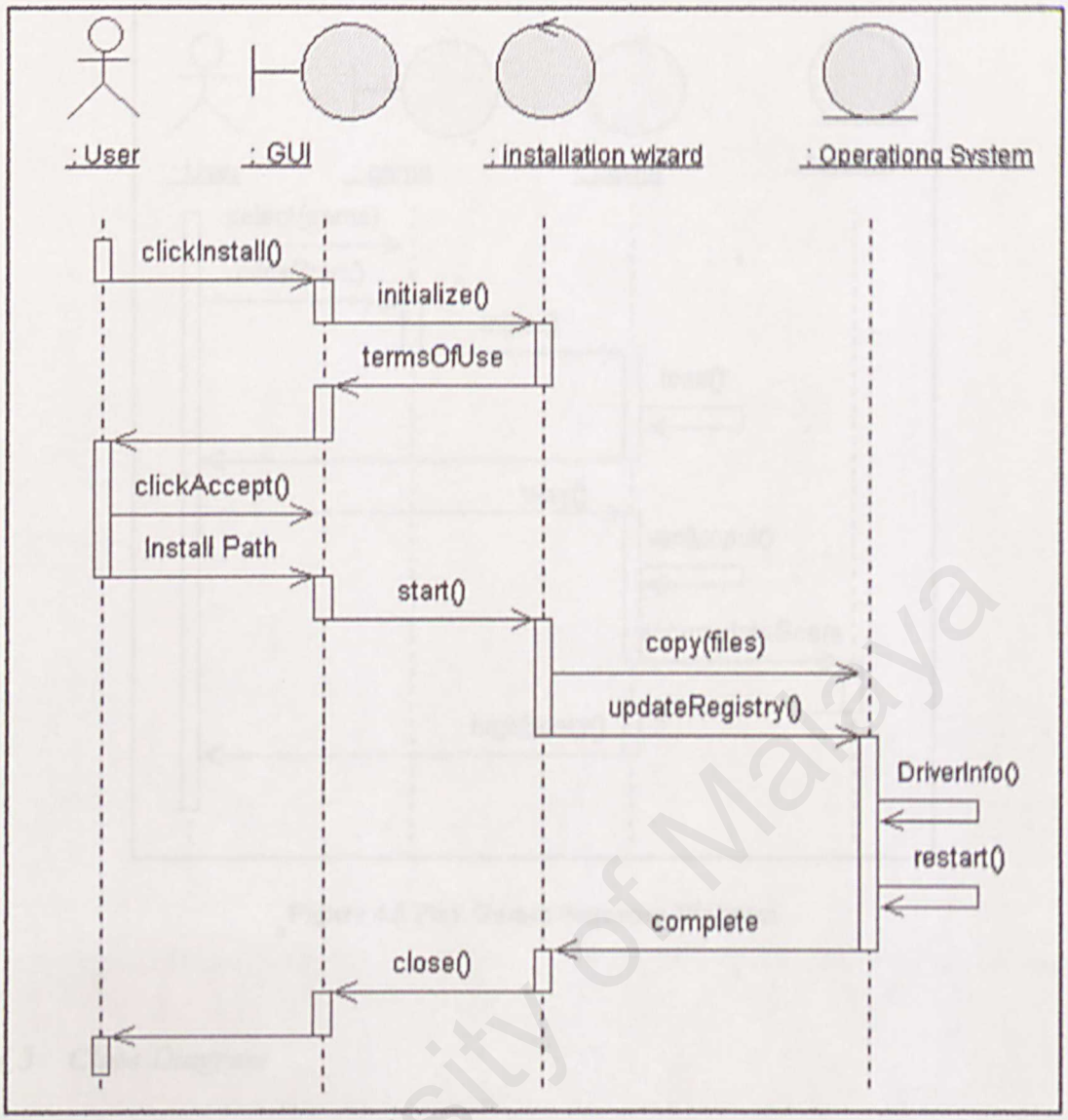

Figure 4.4 Installation Sequence Diagram

Users are required to install the user database and games module before they can continue with quiz or play games. Most of the installation is done by the application wizard and user is only involved in a minor part such as specify the installation location for database and games files.

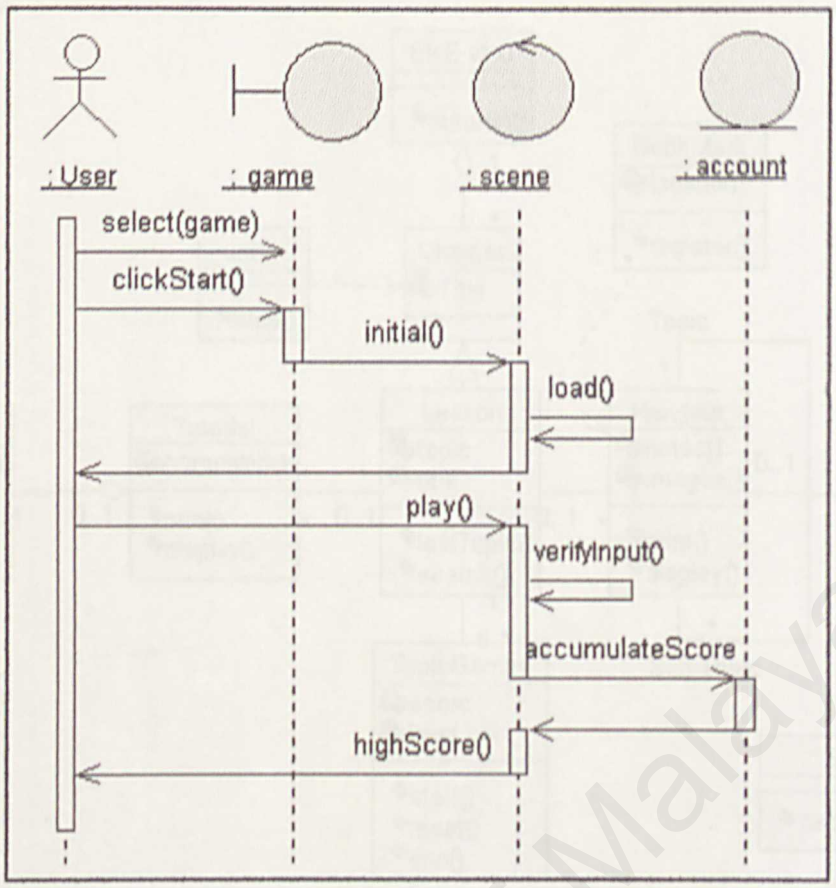

Figure 4.5 Play Games Sequence Diagram

4.4.3 Class Diagram

4.4.3.1 Overall Class Diagram

The class diagram for the overall application captured the attributes and operations for a particular class. User is not included in this class diagram because the user is only manipulate the data via the application such as main menu and no information from he/she is required to perform a particular operation.

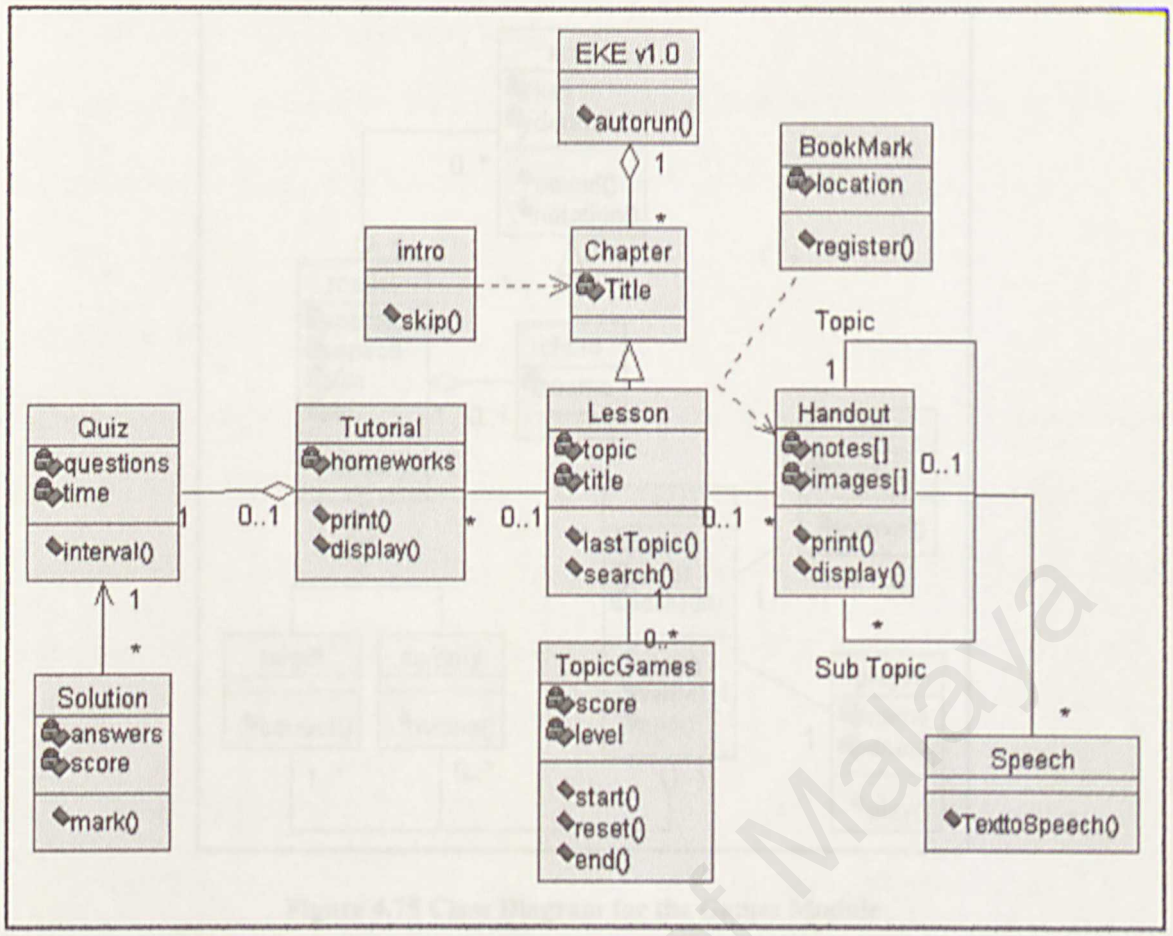

Figure 4.6 Class Diagram for Electronic Keyboard Elementary Multimedia Interactive Package

4.4.3.2 Class diagram for games module

The classes and objects for those two games designed for this learning package is derive from a same class diagram. The reason why only one class diagram was created for two games is simple: both games have the same objects because there are in the same category, and there are different in how to perform the interactive with user. The class model a piece of song, or simply a single or combination of notes or chards or both.

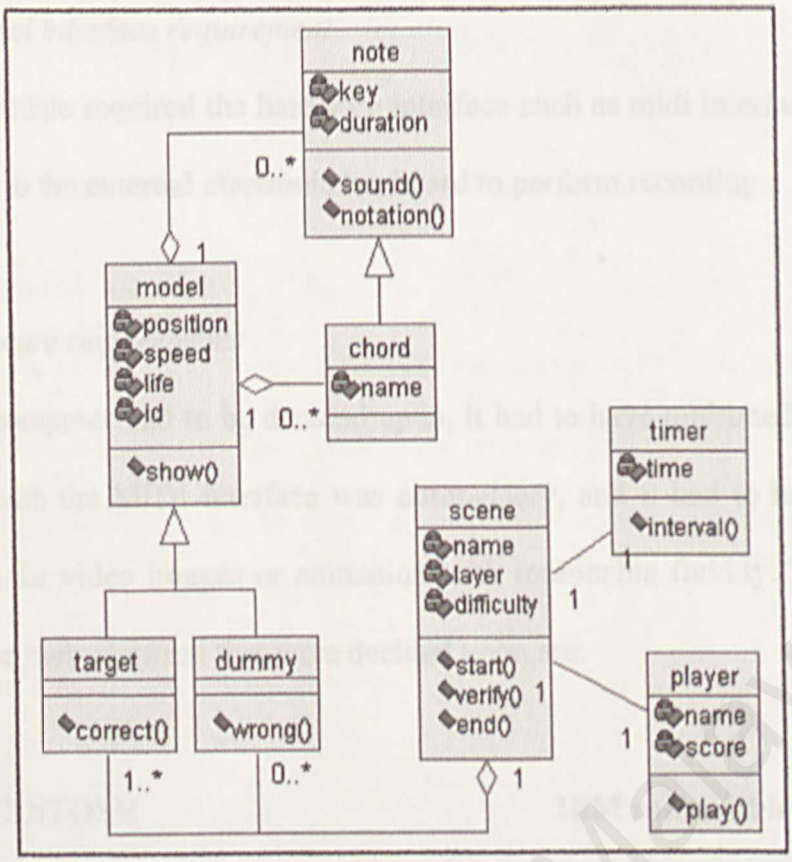

Figure 4.75 Class Diagram for the Games Module

4.5 Non-functional Requirements

A non-functional requirement specifies those aspects relating to performance, accessibility, usability, branding and visual style.

4.5.1 Performance requirements

The courseware is a CD-ROM based application supports in both Microsoft Windows and Macintosh OS X operating system. The teaching lessons should be able to run without prior installation.

4.5.2 External interface requirement

The games module required the hardware interface such as midi interface to connect the computer to the external electronic keyboard to perform recording.

4.5.3 Hardware requirements

A base line computer had to be decided upon, it had to have multimedia capacity, a sound card with the MIDI interface was compulsory, and it had to have sufficient RAM to run the video images or animation with reasonable fluidity. The base line computers for each platform that were decided upon are:

MACINTOSH	**IBM compatible PC**
OS X or above	Window 98 and above
16 MB of RAM	16 MB of RAM
2x CD-ROM	2x CD-ROM

Chapter 5 System Design

5.1 Graphical User Interface Design

The user interface is an agent to interlink between user and the particular information provided in this learning package. Without a well designed interface, the user will feel difficult to access to the contents and so makes the courseware abundant.

5.1.1 Introductions, lessons and handouts

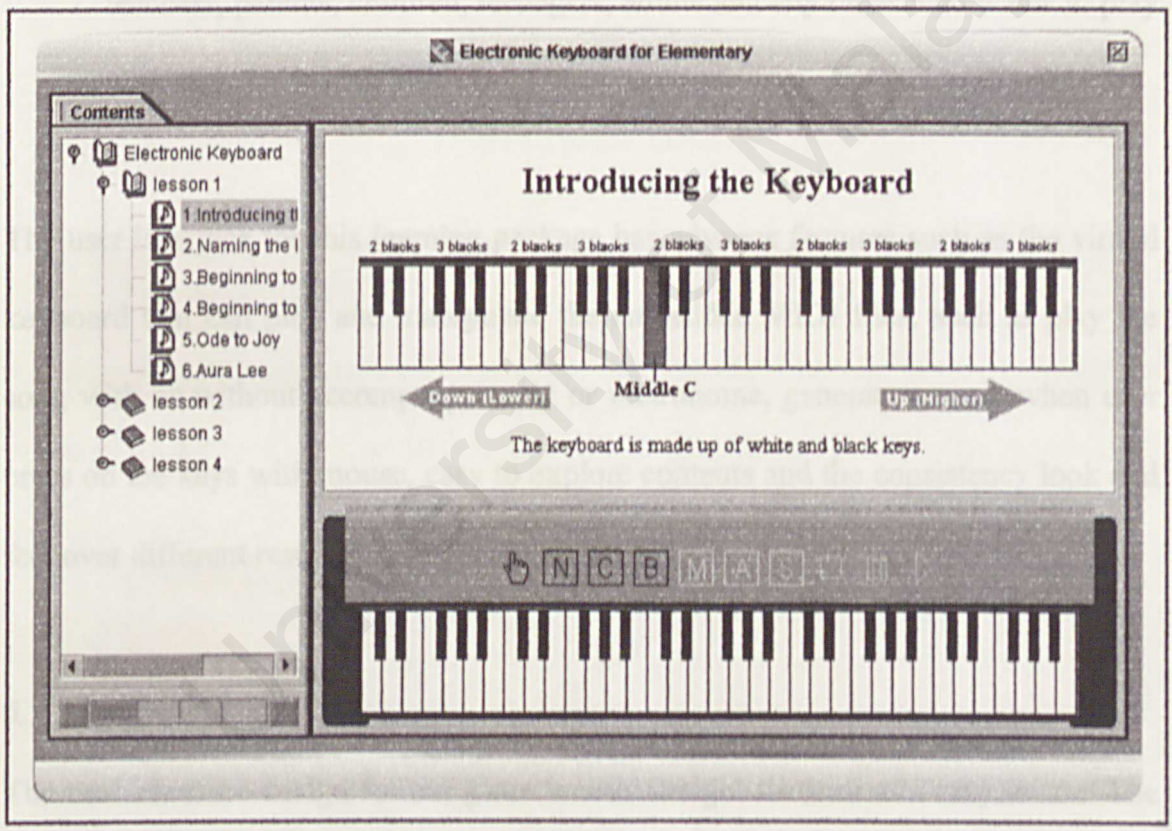

Figure 5.1 User Interface Design for Theory Lessons

This courseware package is designed for all ages users, thus the user interface should provides the attractive looks and feels, designed in easy, clear and straight forward manner.

Basically, the learning package for electronic keyboard provides an understanding icon based buttons and event-driven application. The objectives and motivations to provide the graphical-based application for this learning package are:

- attract user interest, especially for those in primary school to use and explore the courseware
- diminish technique burden, hence user can use easily this courseware without any prior experience or training
- provide an easier and effective learning environment
- build a customizable user interface to suit all layers of users such as piano teachers, parents, children, teenagers, adults and any other who wants to play electronic keyboard

The user interface for this learning package has advance features such as the virtual keyboard that can play and manipulate the embedded MIDI files, such as play the song with or without accompany music or metronome, generate sounds when user press on the keys with mouse, easy to explore contents and the consistency look and feel over different resolutions.

5.1.2 Games

The user interface design for the game is also straight forward and easy to use. The system provided the simple and understanding buttons combination where user will be able to configure the connection and start playing the game within 10 seconds. The background music designed for the different level of difficult has created the good environment to attract user.

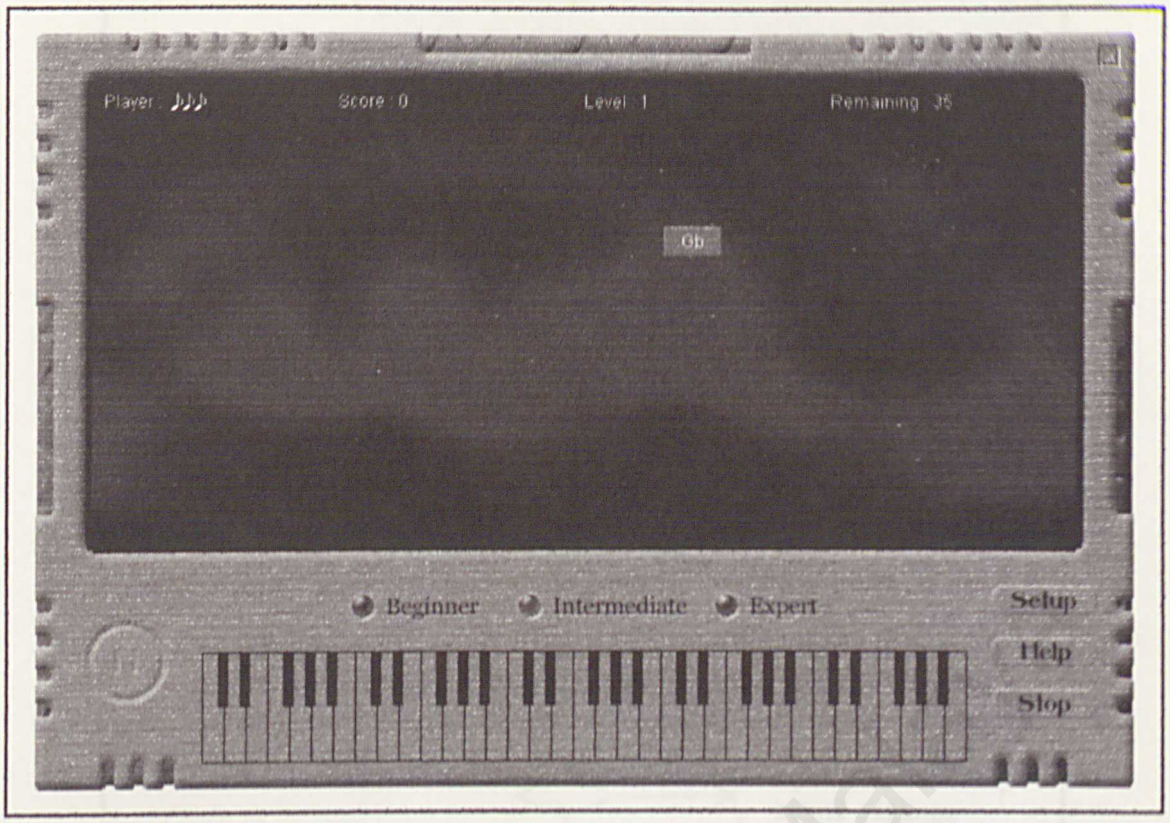

Figure 5.2 Game Module

5.2 Activity Diagram

An activity diagram presents variation of a state diagram. Activity diagram represents performance of actions or subactivities and the transitions are triggered by the completion of the actions or subactivities with the procedure itself. [UML spec]

5.2.1 Learning lessons

The handouts are designed in easy, short and clear HTML files. User can select the topic from the tree panel on the left of the screen; to study the notes, user needs to press the forward or backward button below the tree panel. The progress bar between the buttons shows the total and the current handouts contained in the particular topic.

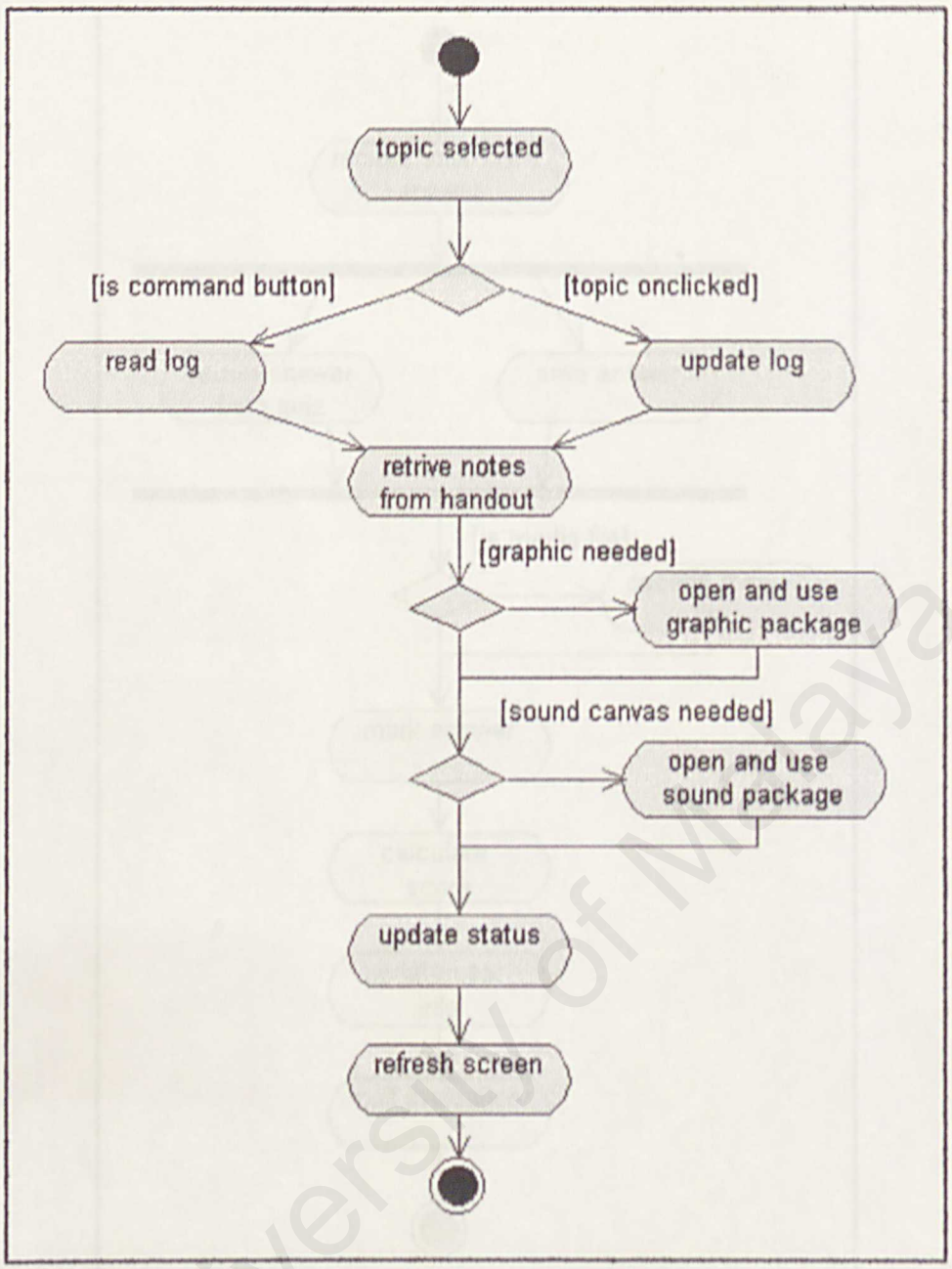

Figure 5.3 Select a Topic to Start

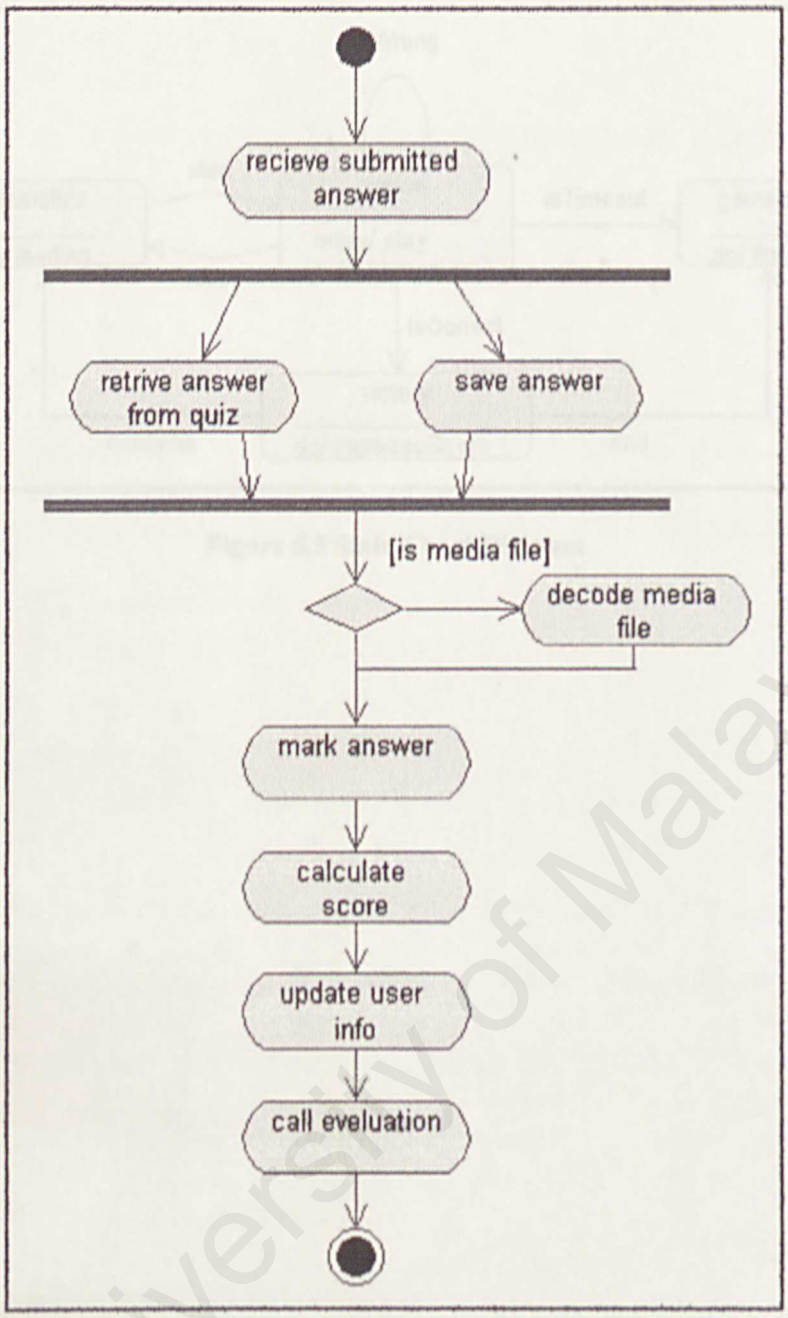

Figure 5.4 Activity Diagram for Game

5.3 State Chart Diagram

State chart diagram is suitable in models the highly interactive application such as game. The design of the game module for this learning package is captured in the following state notation diagram.

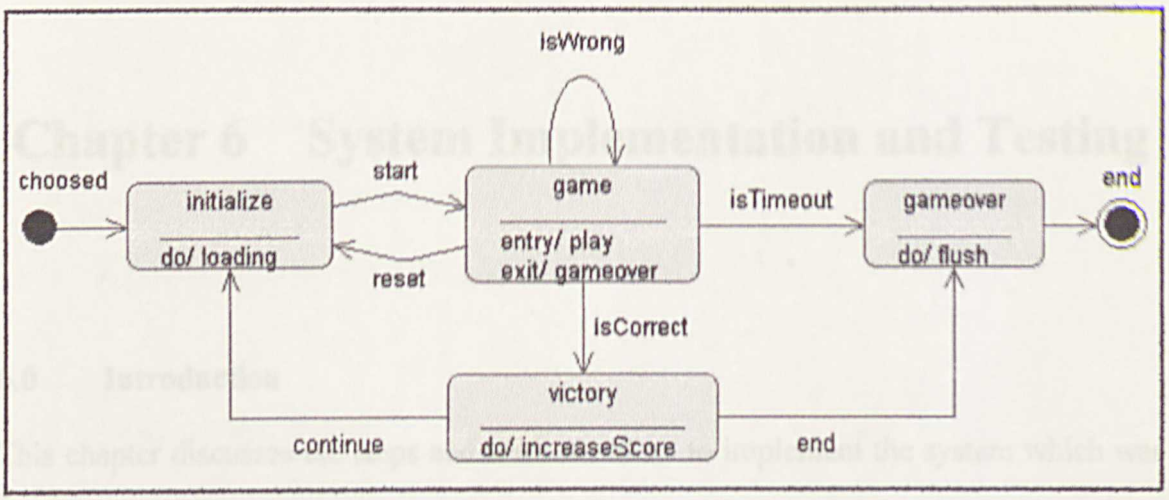

Figure 5.5 State Chart Diagram

Chapter 6 System Implementation and Testing

6.0 Introduction

This chapter discusses the steps and methods taken to implement the system which was designed in the earlier phases. After the implementation, the system will be proceeding to test and check for errors and bugs.

6.1 Implementation

6.1.1 Hardware

Besides the basic hardware and devices required for the Personal Computer architecture such as mouse, keyboard, monitor, processor and RAM, some additional devices and hard such as the real electronic keyboard and sound card with MIDI interface was used to test and build the application for this learning package.

- Panasonic SX-KC211 keyboard
- Creative SB 64 sound blaster
- A MIDI to computer 15 pin port connection cable

6.1.2 Software Tools

The Integrated Development Environments (IDE) designed for build and deploy the Java application not only provided the test editor with syntax checking, but it also embedded the file management system and deployment wizard follows the naming standard suggested by Sun Microsystems.

- Borland JBuilder 8.0 Enterprise Edition Trial version
- Oracle JDeveloper 9.03 review
- Adobe Photoshop 7.0 trial version

6.1.3 Development Strategy

6.1.3.1 Coding Principles

Coding is a process that translates the detail design representation of system into a programming language realization.

The following coding implementations were applied during the implementation of Learning Package for Electronic Keyboard.

i. coding conventions:

Code conventions are important because:

- 80% of the lifetime cost of a piece of software goes to maintenance.
- Hardly any software is maintained for its whole life by the original author.
- Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly.

ii. Readability

There are a number of stylistic elements which can make a program more readable, including the use of horizontal and vertical spacing, the conventions used in

declarations, and etceteras. A good programmer will strive to enhance the visual appearance of the code written. The effort put in will begin to pay dividends during debugging the code:

a. Indentation

Indentation is used to enable a reader to determine the nesting level of a statement at a glance. In order to be useful, indentation must be consistent

- the number of spaces used per indentation level should be between 2 and 5
- the same style of indentation should be used throughout the program.

Proper indentation makes the program much easier to debug.

b. Spaces

Normally in programming the standard for the use of spaces is that you follow ` normal English rules. This means that:

- Most basic symbols in Java (e.g., "=", "+", etc.) should have at least one space before and one space after them, with the following notable exceptions:
  - No space appears before a comma or a semicolon.
  - No space appears before or after a period.
  - No space appears between unary operators and their operands (e.g., "->", "++").

c. Blank Lines

Blank lines should be used to separate long, logically related blocks of code. Specifically:

- In the global section of a compilation unit, the include, const, typedef, and variable declaration sections should be separated by at least one blank line.
- Within a long piece of code, groups of related statements may be separated from other groups by a blank line.
- To be effective as an element of style, blank lines should be used consistently.

iii. Maintainability

a. Use good tools:

Select a good tool such as Borland JBuilder or Oracle JDeveloper is important because the IDE eliminate my worry to worry about editting, compiling, storing, and viewing java code

b. Coding standards:

Use a coding standard: a set of rules for how to package, name, manipulate, and document the classes.

c. No fancy tricks:

Write code to be read by a human, not by a compiler. Not using any optimizations that make the code harder to read or understand.

d. Documentation:

I used javadoc to produce online documentation for every class.

e. Avoid gray areas:

Certain areas of the Java specification can be unclear or buggy. I tried to avoid and do not use the code that depends on a particular interpretation of the release, or that depends on a buggy implementation.

iv. Robustness

In interactive systems safety may be defined differently from the critical systems. Because of the specificity of interactive system architectures, where functional core is generally separated from the user interface, two aspects can distinguish in safety properties: functional core safety and user interface safety.

The security properties that concern the functional core allow ensuring that the functional core primitives execute the required result and that there is no execution conflict. Among the security properties concerning the interface, we can find some that are linked to the system visualization, such as observability, insistence and honesty.

- Observability: The system makes all pertinent information potentially available to the user;
- Insistence: The dialog structure ensures that necessary information is perceived;
- Honesty: The dialog structure ensures that users correctly interpret perceived information;

6.1.3.2 Coding

Method used for coding this learning package is a top-down approach, where high-level modules will be coded first and the lower-level modules will be left, to be filled later. This method allows testing to begin earlier, where some other modules and functions are still being built.

6.2 Testing

Testing is a verification and validation process. Verification refers to the set of activities that ensure that the software correctly implements specific function. It involves using test data and scenarios top verify that each component and the whole work normal and abnormal circumstances.

Software testing is a critical element of software quality and represents the ultimate review of specification, design and coding rules that can serve well as testing objective are:

i. Testing is a process executing a program with the intents of finding an error.

ii. A good data case is the one has high probability of finding an undiscovered error.

iii. A successful test is one that uncovers a yet undiscovered error.

A system is exposed to testing both during the development phase and during the test and integration phase. During the development phase, each function or procedure that is a part if a module is independently develop and thoroughly tested until the entire module is complete. The major different between testing module during its development phase and testing it during the test and integration is that, during the development phase, error will be fixed as they found; during the test on the integration failures, if any, the error will be recorded and the failed module returned to the development process to correct the error.

6.2.1 Testing process

Testing was conducted through the development of this courseware. The testing process consists of unit testing, integration testing and system testing.

i. unit testing

Basically, quality software relied as testing each function, modules. The practice called unit testing, which effective is extremely time consuming and labor-intensive. Using the details design description as a guide,

impotent control path are tested to uncover errors within the boundary of the module. The relative complexity of test and error detected as a result is limited by the constrained scope established for unit testing. Unit test is also referred to as module testing, and is usually performed by the system developer.

ii. integration testing

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. Integration testing identifies problems that occur when units are combined. By using a test plan to test each unit and ensure the viability of each before combining units, I will know that any errors discovered when combining units are likely related to the interface between units. This method reduces the number of possibilities to a far simpler level of analysis.

I did the integration testing the bottom-up strategies:

The bottom-up approach requires the lowest-level units be tested and integrated first. These units are frequently referred to as utility modules. By using this approach, utility modules are tested early in the development process and the need for stubs is minimized. The downside, however, is that the need for drivers complicates test management and high-level logic and data flow are tested late. Like the top-down

approach, the bottom-up approach also provides poor support for early release of limited functionality.

Table 6.1 Sample Integration Test Case for Game

Scenario	Problems	Solutions
User pressed the Setup button while playing the game	Game continue and caused user to lose mark	Pause the game when user pressed the Setup button.
User select the different mode of difficulty while playing	Force the game to proceed to selected mode carried the scores and level	Restart the game
User press the Pause button while the game was stopped or game over	Unexpected action such as the user can continue the game with the scores and level	Group the Play, Pause, Restart and Stop buttons. Set the unrelated button to invisible to prevent user pressed the wrong button
User selected the online help while game was stopped	The online help is not working	Restart the thread
The 36 bricks has been fallen	The game will proceed to next level	Add an algorithm to check whether user has shot all the bricks

iii. System testing

System testing is a series of different tests designed to fully exercise the software system to uncover its limitations and measure its capabilities. The objective is to test an integrated system and verify that it meets specified requirements.

There are two types of system testing that are worthwhile for this courseware.

- Performance Testing

  The purpose of this testing is to test the run-time performance of software within the context of an integrated system. This will show the actual outcome which is the message received and compare to the expected outcome.

6.3 Conclusion

Overall, the system runs smoothly. The system has been tested only by friends and it response accordingly. However, it is not able to perform in the real environment such as let the kids and primary students to play due to the time and resources constraints. As a conclusion, all the objectives have been achieved.

Chapter 7 System Evaluation and Conclusion

7.1 System Strengths

7.1.1 Platform independent

The Microsoft Visual Basic can only build the application operates in the windows base operation system such as Windows ME, Windows 2000 or Windows XP, but not supported in Macintosh, one of the most popular platform used in the multimedia application. By using the Java, this courseware is able to run in any operating system, as long as Java supported with the same output. The system is not only platform independent, but it is also not affected by the different resolution.

7.1.2 First game that can play with electronic keyboard

The **Music Hunter** is the game to help user memorize the position and name of the notes on the keyboard. This is the only game that allows user to play using the electronic keyboard where they can have more rigid experience with the real device.

7.1.3 Simple and user friendly interface

The system designed with the comprehensive and exception handling buttons and input with nice looking to attract user. User does not need to worry if

they pressed the wrong button, this system will crash or effect their operation system and files.

7.1.4 Easy to navigate

The system divides the screen into three blocks where the corresponding contents will display appropriately. User can easily expand and click on the tree to select the handout, read the notes, play the music files or manipulate virtual keyboard to experience playing the electronic keyboard within the same window.

7.1.5 Minimal installation and memory usage

Only Java Runtime Environment 1.4.0 or above is required as this learning package is build on Java. No hard disk space is required to run the program, all handouts and programs can fully run on the machine without prior installation.

7.2 System Constraints

7.2.1 No evaluation and top scores

The learning package for electronic keyboard is not comes with the memory. The handout could not been bookmark and no database was designed to store the players' scores for further evaluation and analysis purpose.

7.2.2 No quiz

No quiz or test was designed to evaluate user's understanding because the limitation of resources and appropriate material suitable for all ages.

7.3 Problems and solutions

7.3.1 Difficulty in Choosing Development Technology and Tools

There are many authoring tools and programming tools in the market are used to develop the multimedia education package. Choosing an appropriate authoring tools and programming tools is a critical process as all the tools has its strength and weakness.

Java is designed for multimedia and platform independent application. It is suitable to develop the complete system such as game and interactive multimedia learning package, moreover has build in package to support multithreading, hardware programming and MIDI system. However, Java is new and the resources are limited, and programming object oriented with pure Java required good understanding both language and objected oriented concept. Furthermore, multimedia programming with Java required the complete system design to overcome the time constrain toward the success of the system.

7.3.2 Determine the Scope of System To Build

It is impossible to develop a full scale complete system for learning electronic keyboard due to the time constraint and resources given. Therefore, a tough decision was made to perform all the topics by putting the short notes and animations in the HTML files.

7.3.3 Lack of music knowledge

To develop a comprehensive learning package for electronic keyboard, student required a deep understanding in music. However, to catch up within two semesters from a absolute dummy to a piano teacher is not possible, so the materials, handouts and music for the learning package were modified from the existing resources such as internet tutorial and books.

7.3.4 Lack of Prior Experience in Object Oriented and Java Programming

Although the object oriented programming and a Java course has been taught during the last few semesters, but no practice from the courses are related to the interactive and multimedia programming. Furthermore, material and resources such as tutorial and source codes for Java were limited over the internet compared with other programming language or authoring tool such as Microsoft Visual Basic or Macromedia Director.

I have spent almost two semesters to do the study and research to overcome a lot of problems such as customize the look and feel, double buffering and animation for games, Java tree, event handling, exception handling and so on.

7.4 Future Enhancements

System development is a dynamic process and changes must be expected. For this learning package, there can still have some improvement to build the better product.

7.4.1 *Evaluation*

Test and quiz are important to test user's understanding. Test is one of the most important elements in the learning process; therefore, adding the test module is critical to complete the learning package. However, the content and difficulty of the question must select carefully to make sure the infant user can understand, and the adult user will not feel bored.

7.4.2 *Database*

Design the light-weight database to store user's scores and answer for the test and game in the user's machine is also important to evaluate and analysis the improvement and understanding of user. The database should be the simple XML or ASCII text file with simple encryption and decryption protection to prevent user alter the data, while no installation required minimizing the disk space usage and providing the platform independent feature.

7.4.3 *More interactive*

The overall performance of the learning package can improve by adding more interactive features such as text to speech synthesizing, more attractive animation and pictures, user customizable theme and background images.

7.4.4 *More Games*

Game is the most effective part in the learning process, especially for the kids. More game should be designed for different topics and concepts to improve

the learning process and catch the user's attention. For example, a game can designed to display the question in grand staff notation and required user to play the correct notes using the keyboard.

7.5 Conclusion

Overall, the learning package is no complete because the absent of some important elements such as the test module and database. However, with the existing object oriented game engine and system architecture, the enhancement and maintenance is faster and easier in the future. New component can directly integrated into the system while the modification of the existing component will not affects other component because the entire system was design in highly coherent and loosely coupled manner.

References

[1] Vliet, Hans Van (2000) *Software Engineering Principle and Practice*. John Wiley & Sons, Ltd.

[2] Eliëns, Anton (1995) *Principals of Abject-Oriented Software Development.* Addison-Wesley Publishers Ltd.

[3] Reynolds, Angus and I. Iwinski, Thomas (1996) *Multimedia Training Develop Technology Based System.* McGraw-Hill.

[4] Marshall, Robert L. (1994) *Eighteenth-Century Keyboard Music.* Schimer Books

[5] J. Bronson, Gary (2000) *A First Book of Visual C++.* Brooks/Cole

[6] Thayer, Rob et al. (1999) *Visual Basic 6 Unleash, Professional Reference Edition.* Sams Publishing

[7] Sun Microsystems Corp. (2002) *The Source for Java™ Technology.* http://java.sun.com/ last updated September-5-2002

[8] Chris Vargas *Intro to Multimedia in Scientific Presentations.* http://tuts.cit.nih.gov/multimedia_vargas/

[9] Hall, Devra (1995) *The CD-ROM Revolution.* Prima Publishing

[10] Buick, Peter (1996) *Live Sound Pa For Performing Musicians.* PC Publishing

[11] *What is MIDI?* www.borg.com/

[12] Harris, Tom *How File Compression Works.* http://www.howstuffworks.com/file-compression2.htm

[13] Jeff Tyson (2001) *How Encryption Works.* http://www.howstuffworks.com/encryption.htm

[14] Adobe System Incorporated (2002) *Photoshop overview.* http://www.adobe.com/products/photoshop/pdfs/overview.pdf

[15] Pellerin, David (1977) The Microsoft Win32 Programming Model: A Primer for Embedded Software Developers. *January 2001 Release of MSDN Library.* Microsoft Corporation

[16] Microsoft Corporation. (2002) *About Microsoft COM.* http://www.microsoft.com/com/about.asp

[17] Blaszcazk, Mike (1999) *Professor MFC with Visual C++ 6.* Wrox Press

[18] Coelho, Rohan and Hawash, Maher (1998) *Direct X, RDX, RSX, and MMX Technology, A Jumpstart Guide to High Performance APIs.* Intel Corporation

[19] Sun Microsystems Inc. (2002) *JDBC Data Access API, The JDBC API Universal Data Access for the Enterprise.* http://java.sun.com/products/jdbc/overview.html last updated Jun-4-2002

[20] PC Magazine (2002) *MySQL review.* http://www.pcmag.com/article2/0,4149,11807,00.asp

[21] Scott, Kendall (2001) *UML Explained.* Addoson-Wesley

[22] Object Management Group Inc. (2002) *Unified Modeling Language (UML), version 1.4.* http://www.omg.org/technology/documents/formal/uml.htm Last Updated Monday, January 21, 2002

[23] Object Management Group, Inc. (2002) *Introduction to OMG's Unified Modeling Language™ (UML™).* http://www.omg.org/gettingstarted/what_is_uml.htm Last Updated Thursday, August 22, 2002

Bibliography

1. Tan Jih Hair, session 2001/2002, *Computer Aided Learning for Chinese Phonetics.* WKES 318/3182 Final Report, Faculty of Science Computer and Information Technology, University of Malaya.
2. Apple Computer, Inc. (1992) *Inside Macintosh.* Addison-Wesley Publishing Company.
3. Schmuller, Joseph (1999) *Sams Teach Yourself UML ion 24 Hours.* Sams Publishing.

Appendix A: User Manual

1.0 Introduction

The Learning Package for Electronic Keyboard is a stand alone platform independent system, which is deploys on the CD-ROM.

This system provided a user friendly, interactive and straight forward learning environment for beginner who interested in pick up the basic skills to play electronic keyboard.

The Learning Package for Electronic Keyboard is divided into two major modules:

- Module 1 – Learning and handouts
- Module 2 – Game

2.0 System requirement

i. Basic Requirement

- Sun Java Runtime Environment (J2RE) 1.4.0 or above

ii. Optional Requirement

- Soundcard with MIDI interface
- MIDI to computer connection cable
- Electronic Keyboard

3.0 How to install Java Runtime Environment

J2RE can download from Sun: http://java.sun.com/j2se/1.4.1/download.html. After download the J2RE, execute the downloaded file and the installation wizard will guide you through the process.

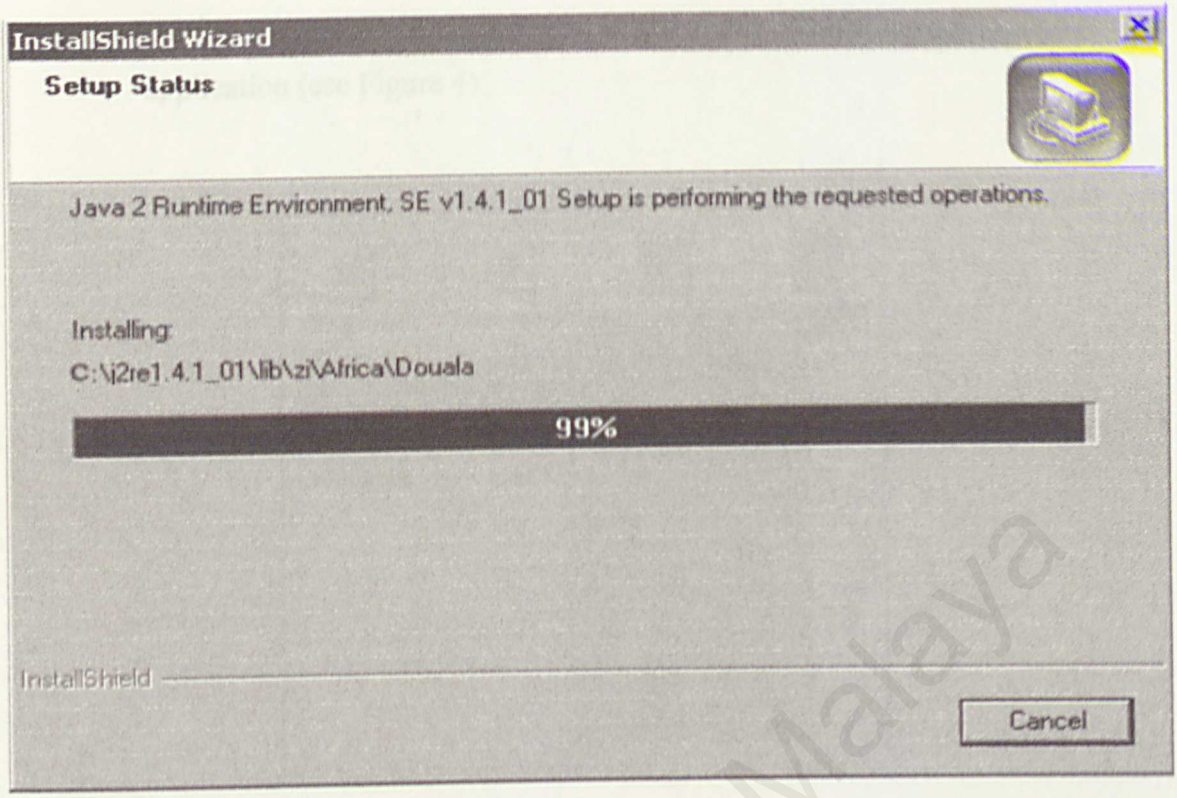

Figure 1 Setting up J2RE 1.4.1

4.0 How to run the Learning and Handouts Module

If you have not install Java Runtime Environment yet, please refer to section 3.0.

4.1 Start the application

To start the application, simply execute the file lessonsUI.jar in the CD-ROM using javaw.exe. To do so in window, assuming D:\ is the letter of CD-ROM drive, and the J2RE was installed in C:\j2re1.4.1_01.

i. Right click on the Start, open Explore, type D:\ on the address input box.

ii. Right click on the lessonsUI.jar and select Open With → Choose Program… (see Figure 2 Choose Program…)

iii. In the Open With dialog box, click Other… and look in to C:\j2re1.4.1_01\bin and select javaw.exe (see Figure 3 Open With…)

iv. Click open and finally click OK in the Open With dialog box to start the application (see Figure 4)

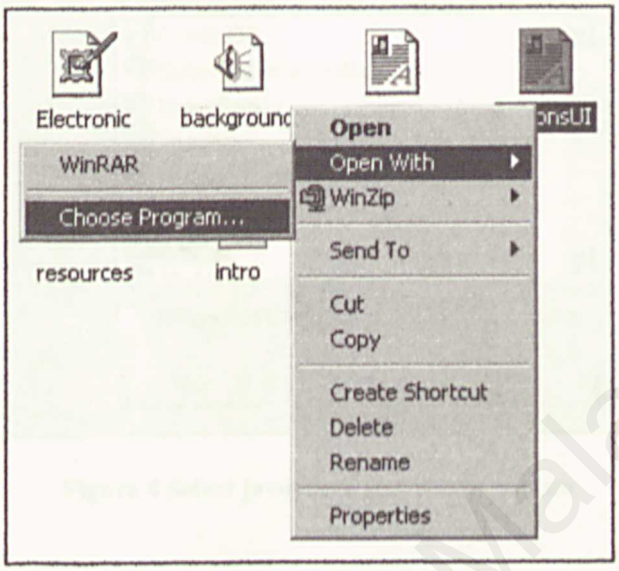

Figure 2 Choose Program...

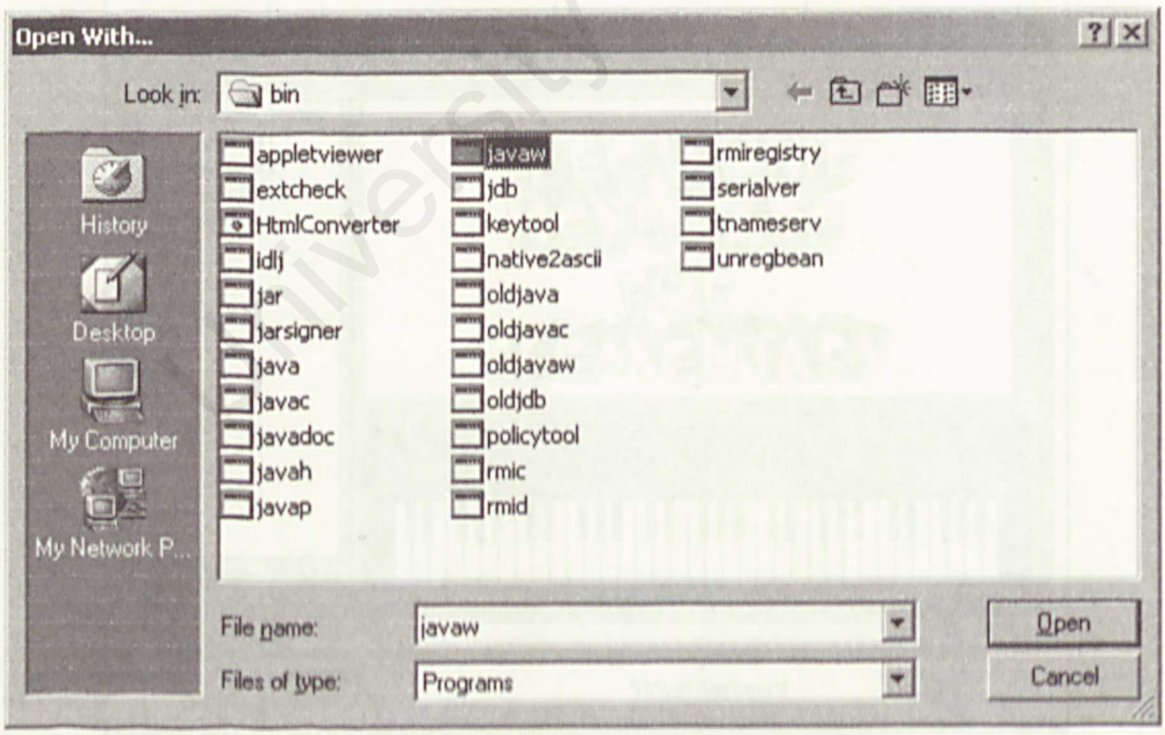

Figure 3 Open With...

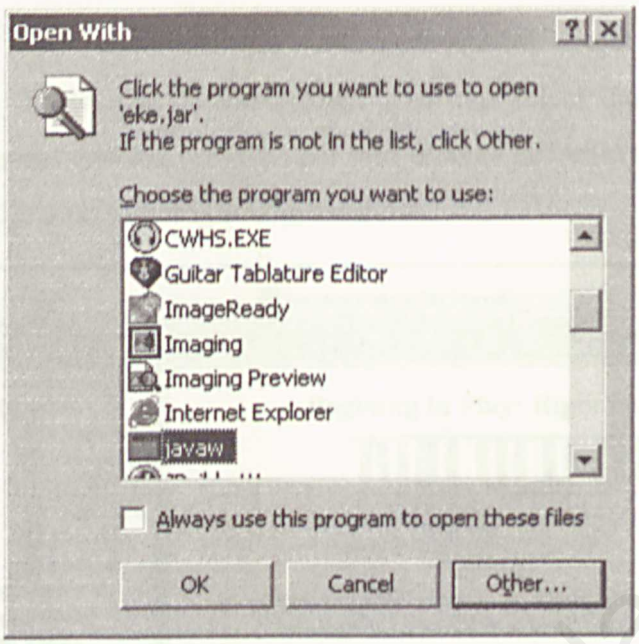

Figure 4 Select javaw.exe and Ready to Run

4.2 Using the Application

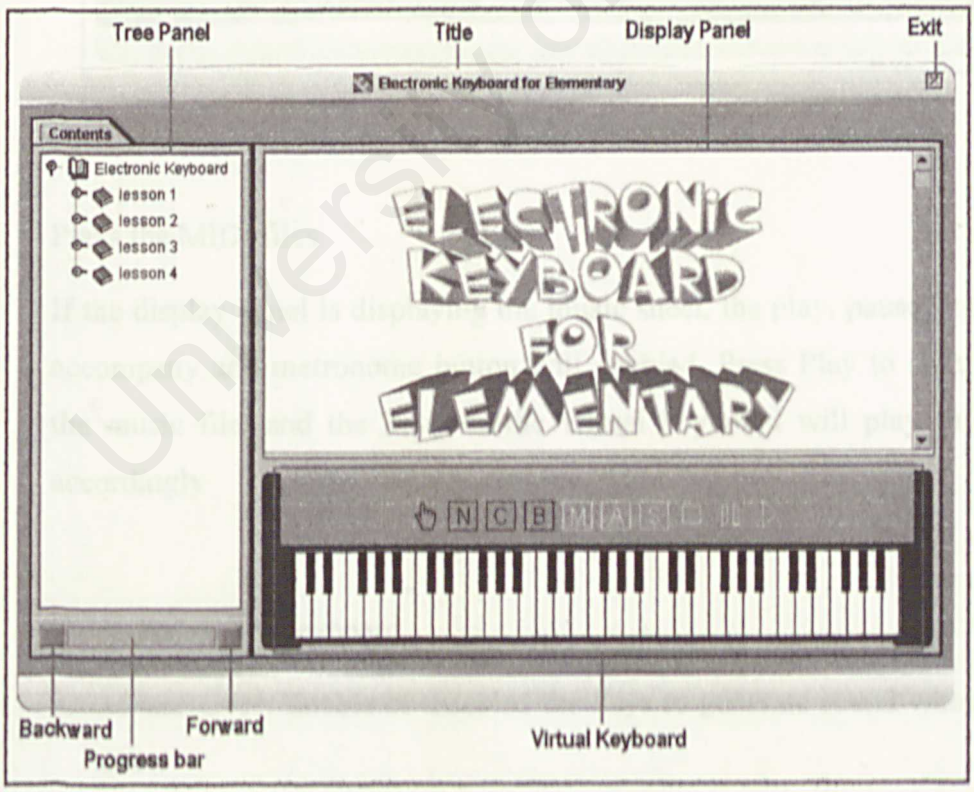

Figure 5 The LesonsUI interface

i. Study the notes

Click on the lessons to expand the tree and select the desired topic. The Display panel and the progress bar will change accordingly. Press forward or backward button to navigate the handouts.

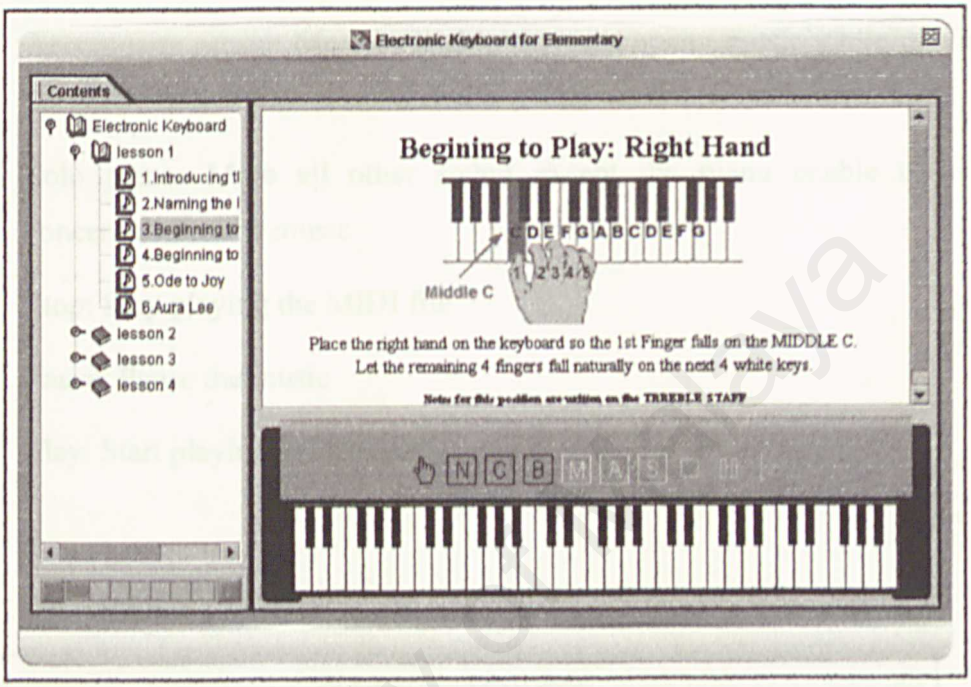

Figure 6 Navigating the handouts

ii. Plays the MIDI files

If the display panel is displaying the music sheet, the play, pause, stop, solo, accompany and metronome button will enabled. Press Play to start playing the music file, and the keys on the virtual keyboard will play (highlight) accordingly

iii. Using the virtual keyboard

a. Mouse Over: Enable or disabled the keys to generate sound when mouse over.

b. Name the Keys: Display the name of the keys on the virtual keyboard

c. Show Middle C: Display the position of the Middle C on the virtual keyboard

d. Background Music: Start or stop the background music

e. Metronome: Mute or play the metronome while playing the MIDI file

f. Accompany music: Mute or enable the accompany music while playing the MIDI file

g. Solo piano: Mute all other sound except the piano enable user to concentrate on the music

h. Stop: Stop playing the MIDI file

i. Pause: Pause the music

j. Play: Start playing the MIDI file

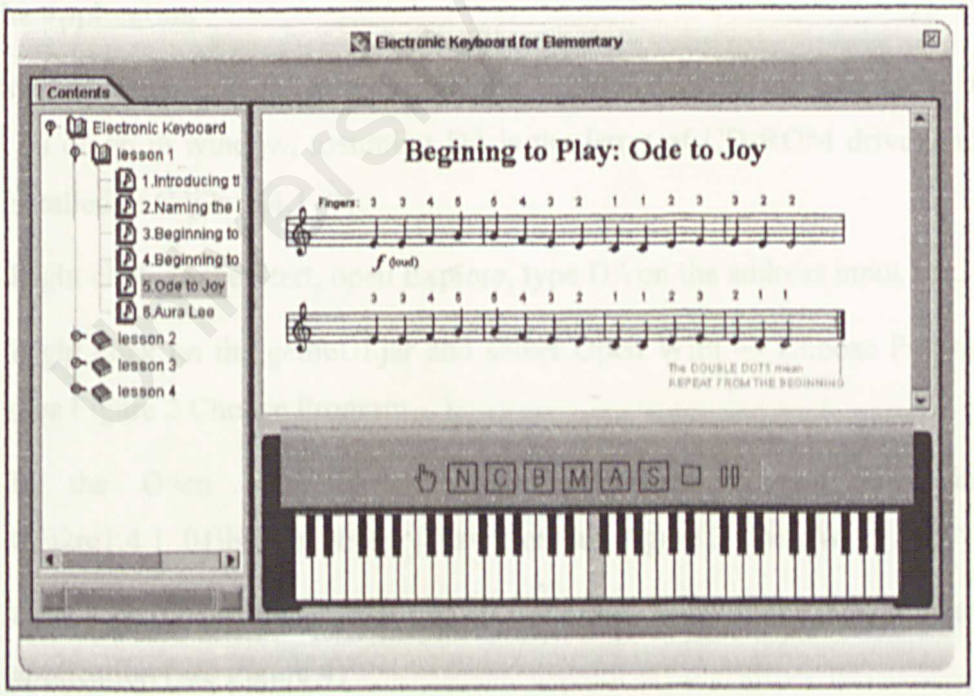

Figure 7 Playing the MIDI file

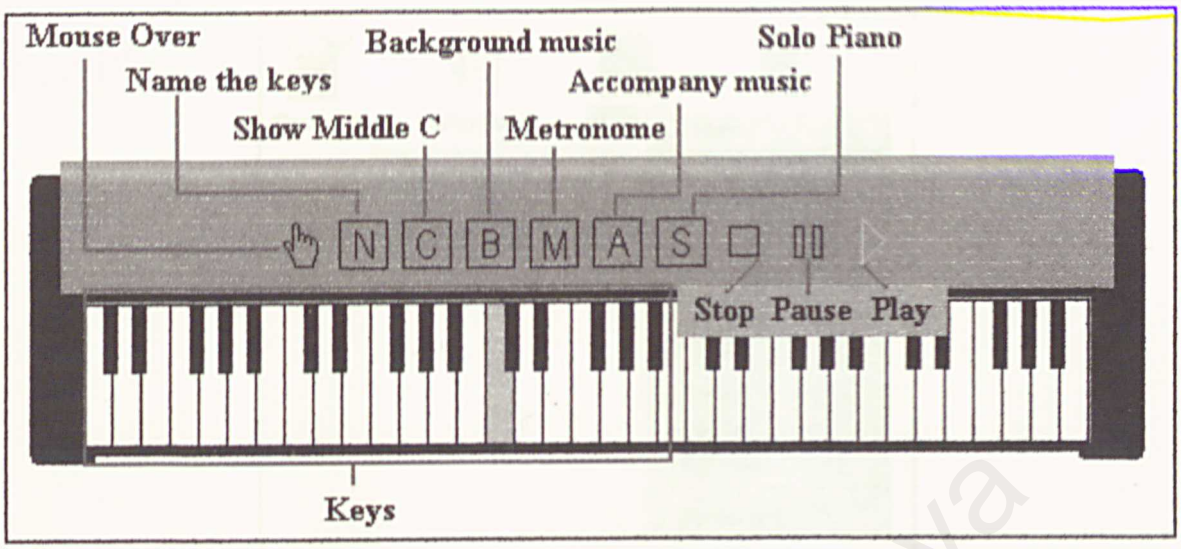

Figure 8 The Virtual Keyboard

5.0 How to run the Game Module

If you have not install Java Runtime Environment yet, please refer to section 3.0.

5.1 Start the application

To start the application, simply execute the file lessonsUI.jar in the CD-ROM using javaw.exe. To do so in window, assuming D:\ is the letter of CD-ROM drive, and the J2RE was installed in C:\j2re1.4.1_01.

v. Right click on the Start, open Explore, type D:\ on the address input box.

vi. Right click on the gemeUI.jar and select Open With → Choose Program… (see Figure 2 Choose Program…)

vii. In the Open With dialog box, click Other… and look in to C:\j2re1.4.1_01\bin and select javaw.exe (see Figure 3 Open With…)

viii. Click open and finally click OK in the Open With dialog box to start the application (see Figure 4)

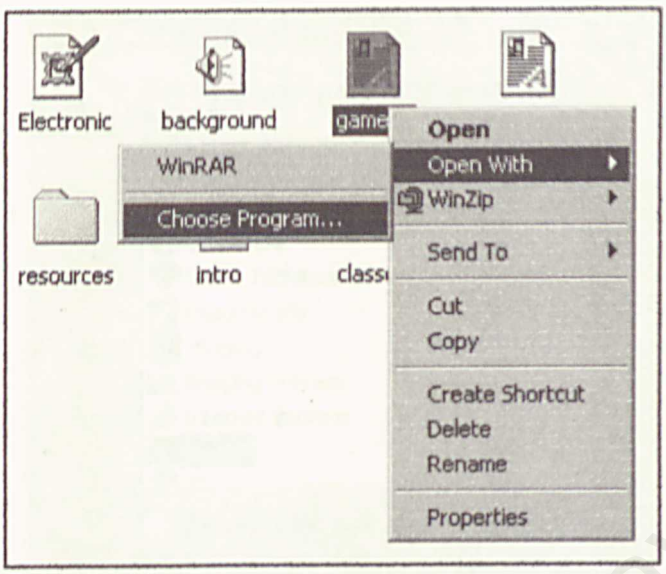

Figure 9 Choose Program...

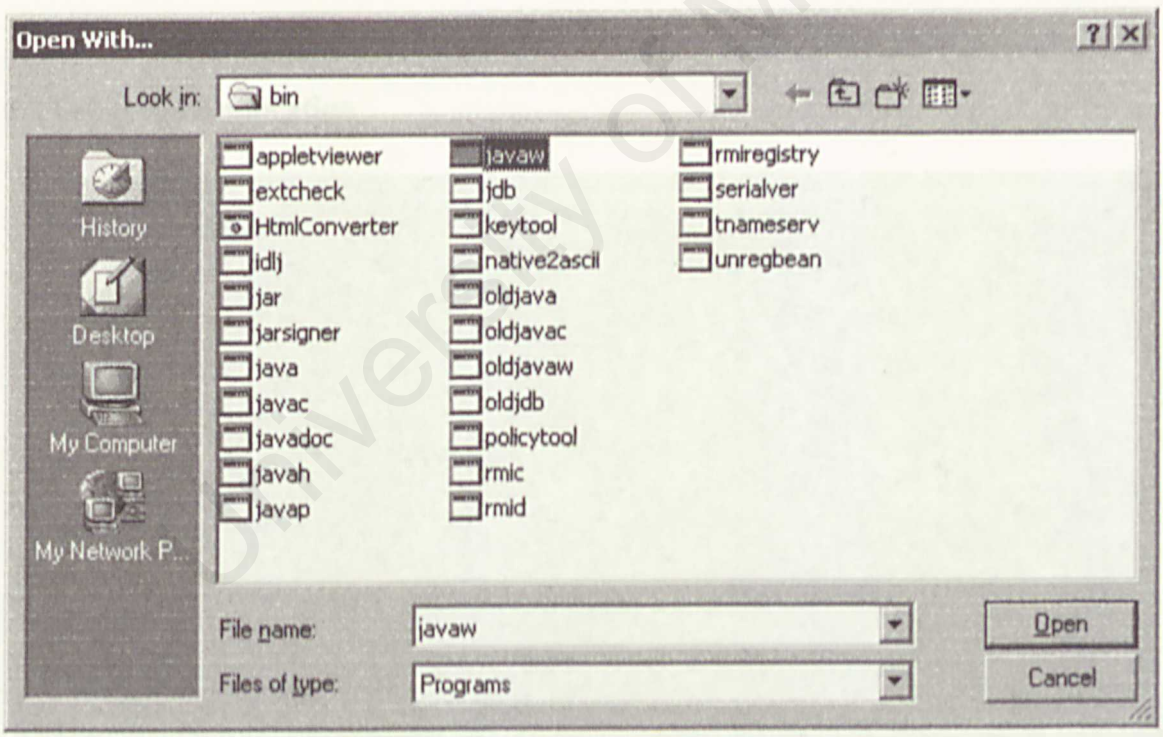

Figure 10 Open With...

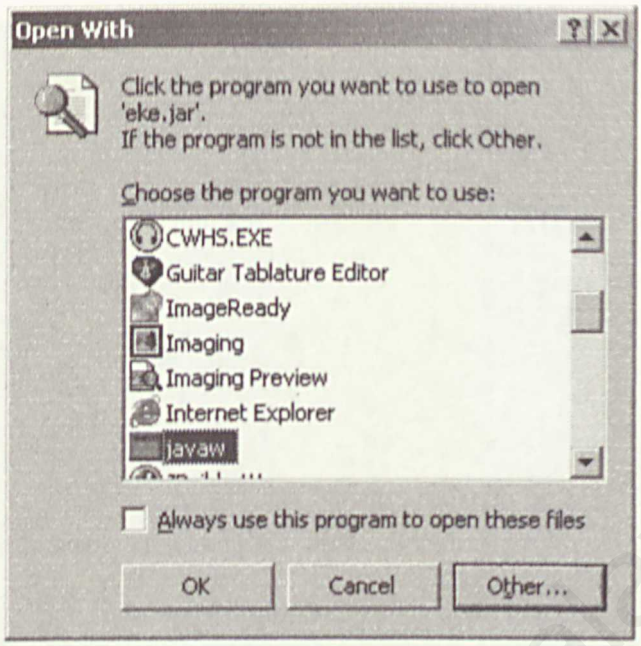

Figure 11 Select javaw.exe and the game is ready to start

5.2 Using the Application

Figure 12 The Music Hunter

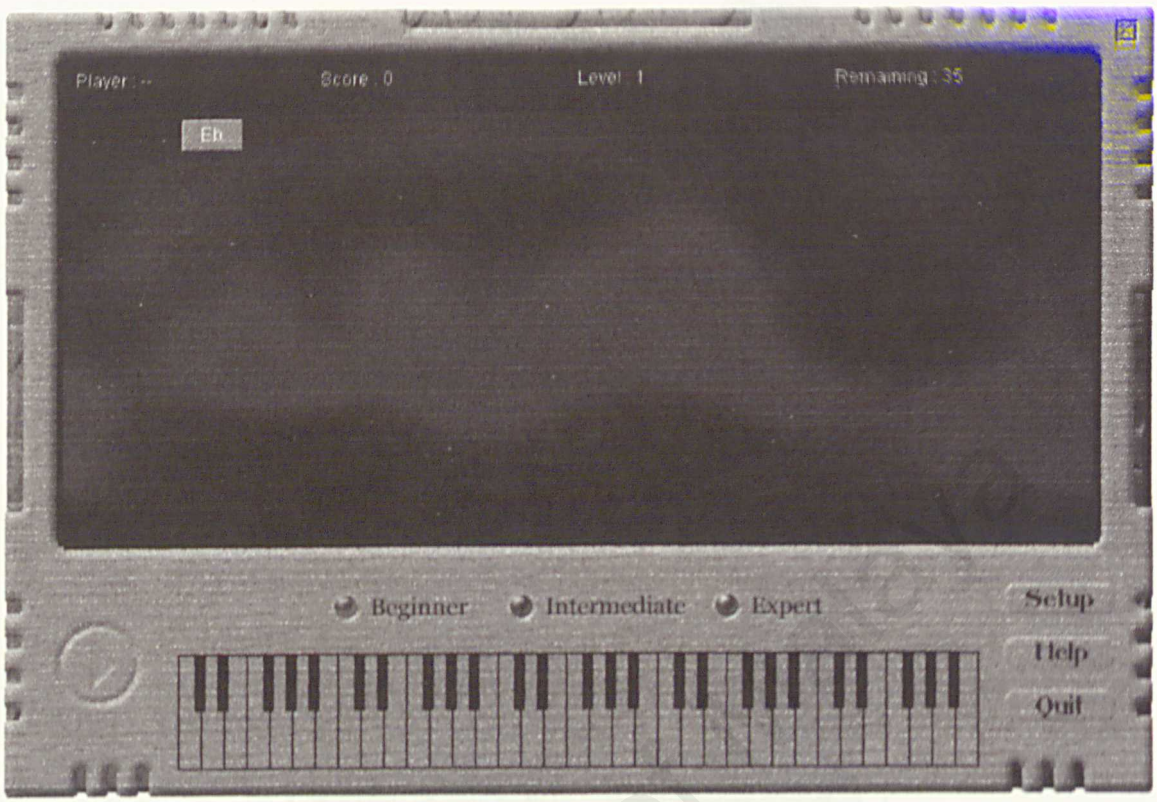

Figure 13 The Game in Action

i. The music hunter is a music brick game. The objective of the game is to eliminate the bricks by press on the correct keys before the bricks fell on the floor.

ii. Three modes were designed for user with different level of expertise to play accordingly. To start the game, user can press on the play button ( ). If user selected the different mode of expertise during the game, the game will be restarted in the new mode.

iii. To pause the game, press on the pause button ( ). The pause button will only appear when the game was stated.

iv. After user paused the game, to resume, user can simply press again on the play button.

v. To play the different level of expertise, for instance, change to the intermediate level, simple press the intermediate button (Intermediate). Note that when you selected the different level, the game will be restarted automatically without a prompt of confirmation.

vi. To view the rules, select the help button (Help). The game will be pause until and the rules will be displayed until user click on resume ( ) or stop button (Stop). Again, the stop will only appear when the game is running or user was selected the help.

vii. To setup the connection to the electronic keyboard, make sure the electronic keyboard was connected to the soundcard. To open a receiver to listen to the keyboard, select setup button (Setup), then choose the MIDI input port (see Figure 14).

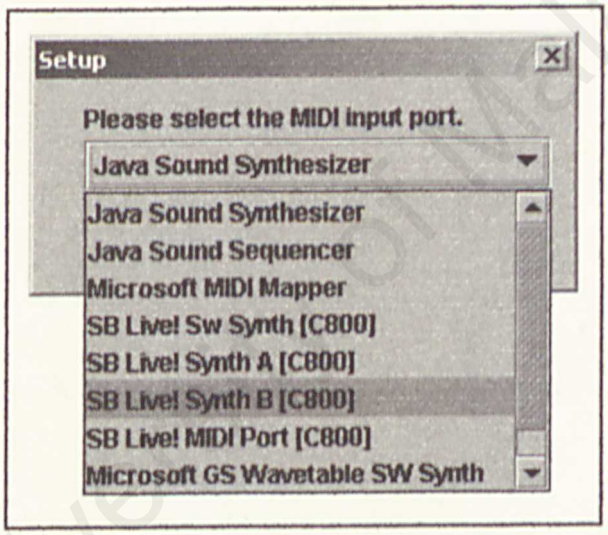

Figure 14 Setting-up the Connection with electronic keyboard